Nonuniform Compressed Sensing Schemes with Sublinear Measurements, Sublinear Time, and Low Entropy

by

Ivan Lau

B.Sc., University of Edinburgh, 2019

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science (Mathematics)

in the
Department of Mathematics
Faculty of Science

SIMON FRASER UNIVERSITY
Summer 2021

Declaration of Committee

Name: **Ivan Lau**

Degree: **Master of Science (Mathematics)**

Thesis title: **Nonuniform Compressed Sensing Schemes with Sublinear Measurements, Sublinear Time, and Low Entropy**

Committee: **Chair:** Weiran Sun
Associate Professor, Mathematics

Jonathan Jedwab
Supervisor
Professor, Mathematics

Ben Adcock
Committee Member
Associate Professor, Mathematics

Paul Tupper
Examiner
Professor, Mathematics

Abstract

Let $\mathbf{x}$ be an unknown vector of length N which has at most k relatively large entries, so that $\mathbf{x}$ is well-approximated by its best k-term approximation $\mathbf{x}_k$. A compressed sensing scheme consists of a measurement matrix $\mathcal{M} \in \mathbb{C}^{m \times N}$ and a recovery algorithm Δ that approximates $\mathbf{x}_k$ as $\widehat{\mathbf{x}} = \Delta(\mathcal{M}, \mathcal{M}\mathbf{x})$. The scheme is considered accurate if, regardless of the unknown vector $\mathbf{x}$, the approximation $\widehat{\mathbf{x}}$ satisfies $\|\mathbf{x} - \widehat{\mathbf{x}}\|_p \leq Ck^{1/p-1/q}\|\mathbf{x} - \mathbf{x}_k\|_q$ with high probability for some absolute constants C and $p \geq q \geq 1$. Other desirable properties are few measurements ($m = O(k \text{ polylog } N)$), fast recovery algorithm ($O(k \text{ polylog } N)$ runtime), and low entropy. In 2014, Iwen presented two compressed sensing schemes: the fastest known accurate scheme, and the only known fast accurate scheme requiring sublinear entropy.

We present a compressed sensing scheme that combines the speed of Iwen's first scheme with the low entropy of Iwen's second scheme, drawing on ideas from disjunct matrix construction due to Porat-Rothschild and Kautz-Singleton. We also give two variants of our scheme. The first variant is deterministic, and has a faster recovery algorithm than all other deterministic accurate schemes. The second variant is measurement-optimal, and requires lower entropy than all other measurement-optimal accurate schemes.

Keywords: Compressed sensing; Sparse recovery; Sublinear-time algorithms; Information theory

Acknowledgements

I am certain that I would not be here if it were not for a number of people whose paths I have been fortunate enough to cross. Most are probably unaware of the extent of the impact their kindness, time, and friendship has had on me. I don't think I can adequately express my gratitude in words, but I will do my best.

First, I would like to thank my supervisor, Jonathan Jedwab for his continuous guidance and support throughout my degree. Jonathan provides me with a lot of freedom and autonomy for my research, and I am particularly grateful for his willingness to supervise me on compressed sensing - a topic that was new to both of us. Theorem-building and problem-solving aside, Jonathan has also taught me various aspects of academic research such as academic writing, presentation skills, journal refereeing, and job applications. On a personal level, Jonathan has been very caring, understanding, and empathetic. When I was going through tough times, he regularly checked in on my well-being and gave me assurance that I could prioritize my relationships, health, and happiness over my thesis.

Next, I would like to thank Mark Iwen from Michigan State University for suggesting a research topic that eventually grew into this thesis, as well as providing helpful feedback on my work. I also benefitted a lot from One World MINDS Seminar, an online seminar Mark co-founded, which contains many excellent talks on mathematical data science and related topics. I would also like to thank my committee member, Ben Adcock for his constructive comments on my progress and work. In addition, Ben provided me with an opportunity to collaborate on a separate research project on group testing with Hooman Zabeti (from School of Computing Science), Leonhardt Unruh and Leonid Chindelevitch (both from Department of Infectious Disease Epidemiology, Imperial College London), and Nick Dexter. This is my first cross-disciplinary collaboration and I have learned a lot from each of them.

It has been a pleasure to be part of Jonathan's research group with Federico Firoozi, Jingzhou Na, Samuel Simon, Shuxing Li, and Thaís Bardini Idalino. My general knowledge and understanding of mathematics and academia life benefitted greatly from our discussions. They were always generous with their time when I pestered them for career advice or feedback on my research work and presentations. Special shout-out to Jingzhou, who made time for my thesis defense preparation even though his thesis deadline was in the same week! Apart from work, our weekly life updates, social gatherings, and conference travels will be something I miss dearly.

I would also like to thank all fellow mathematics graduate students, faculty, and staff. When I was confused by administrative matters, I always knew that I could rely on them (especially Christie Carlson, Nicole Ceron, Anton Iatcenko, and Ahamd Mohktar). Furthermore, when I was on bereavement leave, everyone was very respectful, compassionate, and supportive. In particular, I would like to thank Imin Chen and Petr Lisoněk for granting me many flexible coursework extensions; Justin Gray for being accommodating with my TA workload when I informed about my absence on short notice; Alexandra Wesolek, Daniel Tarnu, and Samantha Zimmerman for covering for me during my period of absence.

The friendships I formed during my time here will also be something I hold dear to my heart. As a non-driver, I am obliged to Jingzhou who often offered me a ride. I also want to thank Jingwen Zhang for making my life in British Columbia much more enjoyable by including me in various dining experiences and travel activities. Jingwen has been remarkably forbearing with Jingzhou and I whenever we selfishly talked about things that only the two of us care about, such as ~~how painful writing with Jonathan is~~ our thesis progress, in a restaurant. Jingwen also taught me various mind-blowing image processing tricks, which were handy when I served as a TA for Computing with Linear Algebra by Petr. I would also like to thank Jingzhou and Jingwen as well as Shuxing and his partner, Jie Jian, for supporting me with accommodation and taking care of me during difficult and uncertain times. Many thanks to other friends who have made my life here more memorable, including but not limited to, Giftson Santhosh, Benjamin Chase, Pengyu Liu, and Qinghong Xu.

I am grateful to Agata Smoktunowicz, He Sun, and Stuart Anderson for their supervision during my undergraduate studies at the University of Edinburgh prior, and for their advice and encouragement with regard to graduate studies. I would also like to thank my final year project collaborators Patrick Kinnear and Dora Puljić for always providing a highly enjoyable working environment; my ACM ICPC teammates Mihai Enache and Tomek Kościuszko for all the fun we had (instead of preparing for the competition); Kyriakos Katsamaktsis for many interesting discussions regarding mathematics, computer science, and travel; my former housemates Albert Wibowo and Danial Harith for planning all the activities that we never did and making me more socially awkward than I already was.

I express my gratitude to my parents and sisters for their care and support throughout all stages of my life. Finally, I would like to thank my partner for her love, support and care.

Table of Contents

List of Tables

List of Figures

List of Algorithms

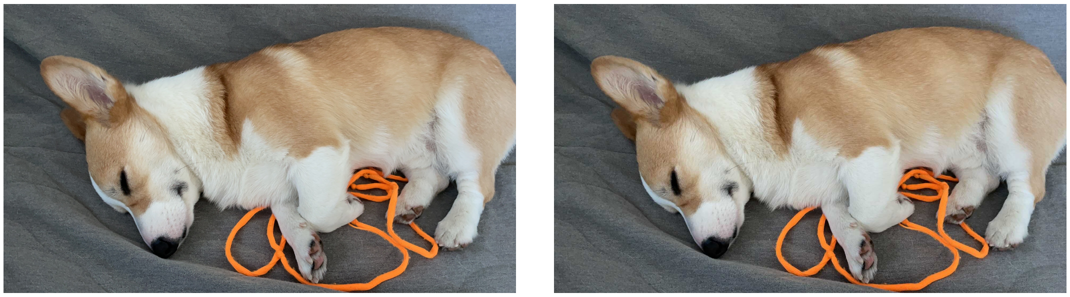
Nomenclature

$[N]$	The set of integers $\{0, 1, \ldots, N-1\}$, see page 10
$\mathbf{x}_k$	A best k-term approximation to $\mathbf{x}$, see page 2
$\widehat{\mathbf{x}}$	Approximation to $\mathbf{x}_k$, see page 3
$\sigma_k(\mathbf{x})_p$	The ℓ_p-error of best k-term approximation to $\mathbf{x}$, see page 4
$\mathbf{x}_S$	Restriction of $\mathbf{x} \in \mathbb{R}^N$ to $S \subseteq [N]$, see page 26
$\mathcal{M}_{\mathrm{C}}$	(K, α)-coherent matrix, see page 19
$\mathcal{M}_{\mathrm{KS}}$	Kautz-Singleton (K, α)-coherent matrix, see page 21
$\mathcal{M}_{\mathrm{PR}}$	Porat-Rothschild (K, α)-coherent matrix, see page 22
$\mathcal{M}_{\mathrm{id}}$	Identification matrix, see page 23
$\mathcal{M}_{\mathrm{est}}$	Estimation matrix, see page 23
$\mathbf{y}_{\mathrm{id}}$	Identification measurement, see page 23
$\mathbf{y}_{\mathrm{est}}$	Estimation measurement, see page 23
$A(n)$	The submatrix of matrix A comprising the rows of A whose entry in column n is 1, see page 30
$A'(n)$	The submatrix of matrix $A(n)$ obtained by deleting column n, see page 30
$\circledast$	The columnwise Kronecker product, see page 37
$\mathcal{B}_N$	The N^{th} bit-test matrix, see page 37

Chapter 1

Introduction

1.1 Motivation



Original image

Compressed image

Figure 1.1: The compressed photo looks similar to the original, even though its size is only 27% of the original photo.

Consider taking a photo using a 12-megapixel camera on a modern phone, which produces a file of size 36MB. When this photo is sent to a friend over a messaging app, the photo that is received is likely to have been automatically compressed to a significantly smaller size. However, it is often difficult to perceive the difference between the compressed photo and the original with the naked eye, as demonstrated in Figure 1.1. The core idea behind this observation is that most of the data are inessential.

To explore this idea in more detail, we say a real-valued signal of length N is k-sparse if at most k of its entries are non-zero, and is k-compressible (or approximately k-sparse) if at most k of its entries are relatively large. In practice, many signals are k-compressible for some $k \ll N$ after an appropriate change of basis. For example, the signal might be produced by a camera making at least N measurements of the RGB intensity at the pixels of the image, and the signal becomes k-compressible under a change of basis using the discrete cosine transform. If $\mathbf{x}$ is k-compressible, then a best k-term approximation to $\mathbf{x}$ (retaining k of its largest absolute entries and zeroing out the other entries) serves as a good compressed version. This is the core idea behind the JPEG compression algorithm.

$$\begin{tikzcd}
\mathbf{s} \arrow[r, "\Psi"] & \mathbf{x} \arrow[d, "\text{best } k\text{-term approximation}"] \\
\widehat{\mathbf{s}} & \mathbf{z} \arrow[l, "\Psi^{-1}"]
\end{tikzcd}$$

Figure 1.2: Signal $\mathbf{s}$ is k-compressible after an appropriate change of basis ($\mathbf{x} = \Psi\mathbf{s}$)

We see that substantial effort is devoted to measuring all entries of the original signal, only for these entries to be discarded in the compressed version. This motivates the questions:

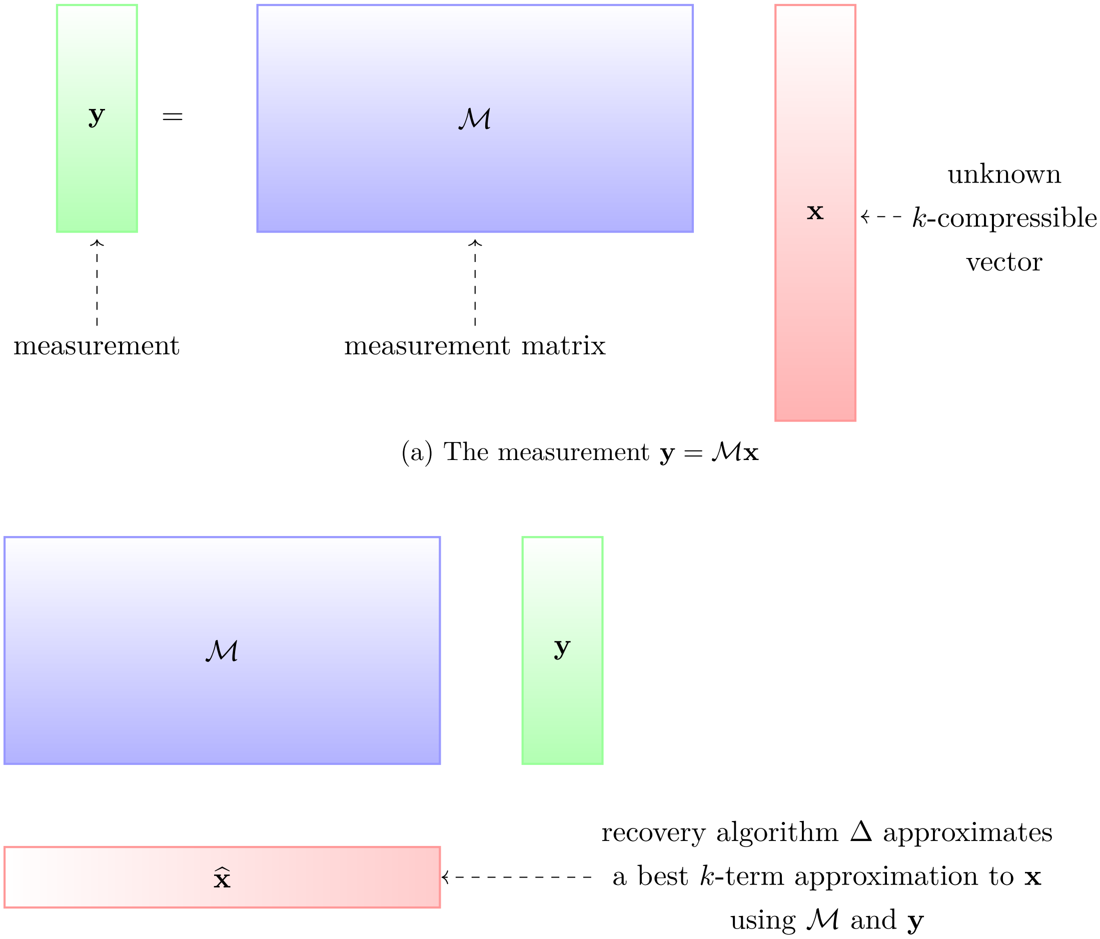
> why go to so much effort to acquire **all** the data when **most** of what we get will be thrown away? Can we not just **directly measure** the part that will not end up being thrown away? (David Donoho [Don06])

An alternative approach, proposed about 15 years ago [CT05, CRT06a, CT06a, CRT06b, Don06] and now known as compressed sensing (or compressive sensing or compressive sampling), is to carry out the measurements and compression simultaneously. That is, we directly acquire the compressed version of the signal, using significantly fewer measurements than the signal length. Compressed sensing has since received much attention from researchers in mathematics, statistics, computer science, and engineering — see [Can06, Bar07, CW08, Rom08, EK12, FR13, Don18] for an overview, as well as [Ric] for an extensive list of literature from 2005 to 2014.

1.2 Principles of Compressed Sensing

Problem formulation. Let $\mathbf{x} \in \mathbb{R}^N$ be an unknown k-compressible signal, and $\mathbf{x}_k$ be a best k-term approximation to $\mathbf{x}$. We may access $\mathbf{x}$ only indirectly by computing $m \ll N$ linear combinations (of our choice) of its entries, and we wish to use these m measurements to produce an approximation to some[1] $\mathbf{x}_k$. In this way, we directly acquire the compressed version of the signal.

Restating this process using matrix notation, we choose a measurement matrix $\mathcal{M} \in \mathbb{C}^{m\times N}$ to compute a measurement $\mathbf{y} = \mathcal{M}\mathbf{x}$, and approximate some $\mathbf{x}_k$ as $\widehat{\mathbf{x}}$ using $\mathcal{M}$ and $\mathcal{M}\mathbf{x}$. We write Δ for the recovery (or decoding) algorithm that calculates $\widehat{\mathbf{x}}$ from $\mathcal{M}$ and $\mathbf{y}$.



(a) The measurement $\mathbf{y} = \mathcal{M}\mathbf{x}$

(b) The recovery: $\widehat{\mathbf{x}} = \Delta(\mathcal{M}, \mathbf{y})$

Figure 1.3: The goal is to design a (distribution for a) matrix $\mathcal{M} \in \mathbb{C}^{m\times N}$ and a recovery algorithm Δ, such that a best k-term approximation to an unknown vector $\mathbf{x} \in \mathbb{R}^N$ can be approximated accurately from the matrix $\mathcal{M}$ and (non-adaptive linear) measurements $\mathcal{M}\mathbf{x}$.

Remark 1.2.1. As noted in Figure 1.2, we may assume $\mathbf{x}$ to be k-compressible after an appropriate change of basis. Let $\mathbf{s} \in \mathbb{R}^N$ be the original signal, and let Ψ be a change-of-basis matrix such that $\mathbf{x} = \Psi\mathbf{s} \in \mathbb{R}^N$ is k-compressible. Then we may calculate $\mathbf{y} = \mathcal{M}\mathbf{x}$

[1] The possibility that $\mathbf{x}_k$ is not unique is largely a technicality. In practice, if we choose the value of k correctly, then the property that $\mathbf{x}$ is k-compressible will make $\mathbf{x}_k$ unique.

as $\mathbf{y} = \mathcal{M}\Psi\mathbf{s}$, regarding $\mathcal{M}\Psi$ as the measurement matrix. After applying the recovery algorithm Δ to obtain $\widehat{\mathbf{x}}$, we obtain a compressed version of $\mathbf{s}$ via $\widehat{\mathbf{s}} = \Psi^{-1}\widehat{\mathbf{x}}$. We shall henceforth work only with $\mathcal{M}$ and $\mathbf{x}$.

We call a matrix-algorithm pair $(\mathcal{M}, \Delta)$ a compressed sensing scheme. The linear measurements $\mathbf{y} = \mathcal{M}\mathbf{x}$ are non-adaptive: the measurement matrix $\mathcal{M}$ is determined in advance, without knowledge of any of the entries of $\mathbf{y}$. A good scheme $(\mathcal{M}, \Delta)$ satisfies the following properties: $\widehat{\mathbf{x}}$ is close to some $\mathbf{x}_k$ (regardless of the value of the unknown vector $\mathbf{x}$), and the number of rows of $\mathcal{M}$ (which is the number of measurements) is small. These properties will be described in more detail below as (P1) and (P2). Depending on the application, we may also desire some other properties such as fast recovery algorithm, low-entropy matrix construction, memory-efficient recovery algorithm, robust recovery algorithm, and fast update/encoding. In this thesis, we shall design a compressed sensing scheme $(\mathcal{M}, \Delta)$ which *simultaneously* satisfies all the following properties (P1)–(P4).

(P1) Mixed (ℓ_p, ℓ_q)-instance optimal. For real $p \geq 1$, define

$$\sigma_k(\mathbf{x})_p := \inf\left\{ \|\mathbf{x} - \mathbf{z}\|_p : \mathbf{z} \in \mathbb{R}^N \text{ is } k\text{-sparse} \right\}.$$

It follows from the definition of a best k-term approximation that each $\mathbf{x}_k$ attains the infimum, and therefore $\sigma_k(\mathbf{x})_p$ is the ℓ_p-error of (each) best k-term approximation to $\mathbf{x}$. We can therefore quantify how accurately $\widehat{\mathbf{x}}$ approximates some $\mathbf{x}_k$ by comparing $\|\mathbf{x} - \widehat{\mathbf{x}}\|_p$ with $\sigma_k(\mathbf{x})_p$: we say a scheme is $\underline{\ell_p\text{-instance optimal of order } k \text{ with constant } C > 0}$ if

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_p \leq C\, \sigma_k(\mathbf{x})_p.$$

A more general formulation [CDD09, §8] allows the use of different norms on the left and right side of the inequality: for $p \geq q \geq 1$, we say a scheme is $\underline{\text{mixed } (\ell_p, \ell_q)\text{-instance optimal of order } k \text{ with constant } C > 0}$ if

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_p \leq \frac{C}{k^{1/q-1/p}}\, \sigma_k(\mathbf{x})_q. \tag{1.1}$$

We shall refer to (1.1) as an ℓ_p/ℓ_q error guarantee.

(P2) Small number of rows. We want the row count m of the measurement matrix $\mathcal{M}$ (namely the number of measurements) to be small. Ideally, we want m to have a growth rate $O(k \text{ polylog} N)$. In many applications, this property is highly desirable. For example, in magnetic resonance imaging (MRI), fewer measurements leads to shorter scan (data acquisition) time. This is important in certain scenarios such as when the patient is not allowed to breathe (for example during a lung scan) or when the patient is a fidgety child. Indeed, current MRI technology based on concepts from compressed sensing

[LDP07, HHL10, MAD+12, LDSP08, FBB+17, Don18] can reduce the scan time by a factor of six or more.

(P3) Fast recovery algorithm. The time taken by the recovery algorithm Δ to compute $\widehat{\mathbf{x}}$ from $\mathcal{M}$ and $\mathbf{y}$ should be small. Ideally, we want the runtime to have a growth rate of $O(k \text{ polylog} N)$. This property is of enormous significance in certain applications of compressed sensing, such as network traffic monitoring, where the signal $\mathbf{x}$ stores the number of times each of the different IP addresses is encountered. In MRI applications, the runtime of the recovery algorithm corresponds to the time needed to reconstruct the image after the scan has been performed.

(P4) Low-entropy matrix construction. We would like to use a small amount of randomness in constructing the measurement matrix $\mathcal{M}$. This is important because measurement implementations requiring a lot of randomness can present considerable engineering challenges, making them infeasible or too expensive to be useful for certain applications (see [DE11] for a more detailed discussion). We can think of the amount of randomness as the average number of bits of information needed to generate a measurement matrix $\mathcal{M}$. To make this precise, we need the following concepts. We view a construction of a matrix $\mathcal{M}$ as a realization of some matrix-valued random variable X. Given a random variable $X\colon \Omega \to E$ (where Ω is a sample space and E is a measurable space), the entropy $H(X)$ of X is defined by

$$H(X) \coloneqq \begin{cases} -\sum_x p(x) \cdot \log_2 p(x) & \text{if } X \text{ is a discrete random variable with probability mass} \\ & \text{function } p(x) \coloneqq \mathbb{P}(X = x) \\ & \\ -\int f(x) \log_2 f(x)\, dx & \text{if } X \text{ is a continuous random variable with probability} \\ & \text{density function } f(x). \end{cases}$$

We want the entropy $H(X)$ associated with the measurement matrix $\mathcal{M}$ to be low, and ideally to have a sublinear growth rate with respect to N. We shall refer to the associated entropy of a measurement matrix/scheme simply as the entropy of a measurement matrix/scheme. In certain settings, we may desire a deterministic measurement matrix construction, that is, one whose (associated) entropy is zero.

Uniform and nonuniform recovery models[2]. When randomness is incorporated in accordance with property (P4), the measurement matrix $\mathcal{M}$ is randomly chosen from some specified distribution. There are two principal probabilistic models for an ℓ_p/ℓ_q guarantee (1.1) to hold: the uniform and the nonuniform recovery model. In the uniform recovery

[2] also known as "for-all" and "for-each" models

model, for a single randomly-chosen matrix $\mathcal{M}$, with high probability the error guarantee is satisfied for all $\mathbf{x} \in \mathbb{R}^N$. For the nonuniform recovery model, for each fixed (unknown) $\mathbf{x} \in \mathbb{R}^N$ and for a matrix $\mathcal{M}$ chosen randomly and independently for each $\mathbf{x}$, with high probability the error guarantee is satisfied. A scheme which satisfies an ℓ_p/ℓ_q guarantee in the uniform recovery model also satisfies the same error guarantee in the nonuniform recovery model (by choosing the same matrix for each $\mathbf{x}$ from a degenerate distribution).

Trade-offs and lower bounds. An important theoretical and practical challenge in the design of compressed sensing schemes is how to achieve good performance across multiple properties simultaneously. When the properties are in conflict, this necessarily involves trade-offs. For example, the scheme that uses the identity matrix as the measurement matrix $\mathcal{M}$ to obtain $\mathbf{y} = \mathcal{M}\mathbf{x} = \mathbf{x}$, and a recovery algorithm $\Delta(\mathcal{M}, \mathbf{y}) = \mathbf{y}$ that outputs $\widehat{\mathbf{x}} = \mathbf{y} = \mathbf{x}$, satisfies (P1), (P3), and (P4) but fails terribly at (P2). Conflicts between properties also occur because of the following information-theoretical lower bound: a scheme which achieves an ℓ_1/ℓ_1, ℓ_2/ℓ_1, or ℓ_2/ℓ_2 error guarantee under the nonuniform recovery model must have a growth rate of $\Omega(k \log(N/k))$ for the number of measurements [BIPW10, PW11], and therefore the same lower bound for the runtime (unless the measurement matrix is trivial). There has been considerable work attempting to meet these lower bounds. We summarize this work and other principal previous results in Table 1.1. The listed schemes have either the best performance for one or more of properties (P1)–(P4), or else strong performance across all four properties.

Paper	**Error guarantee**	**D/U/N**	**Row count of $\mathcal{M}$**	**Runtime of Δ**	**Entropy**
[BDF+11, Mix15]	ℓ_2/ℓ_1	D	$k^{2-\varepsilon}$	LP	0
Our Corollary 3.5.1	ℓ_2/ℓ_1	D	$k^2 \log^2 N$	$k^2 \log^2 N$	0
[KN10]	ℓ_2/ℓ_1	U	$k \log(N/k)$	LP	$O\left(k \log(N/k) \cdot \log\left(k \log\left(N/k\right)\right)\right)$
[GSTV07]	ℓ_2/ℓ_1	U	$k \log^{\geq 2} N$	$k^2 \log^{\geq 2} N$	$\Omega(N)$
[GLPS17]	ℓ_1/ℓ_1	U	$k \log N$	$k^{>1} \log^{\geq 2} N$	$\Omega(N)$
[CM06, Theorem 4]	ℓ_2/ℓ_2	N	$k \log^3 N$	$k \log^3 N$	$\Omega(N)$
[GLPS12, Theorem 1.1]	ℓ_2/ℓ_2	N	$k \log(N/k)$	$k \log^{\geq 2} N$	$\Omega(N)$
[NS19, Theorem 1.2]	ℓ_2/ℓ_2	N	$k \log(N/k)$	$k \log^2(N/k)$	$\Omega(N)$
[Iwe14, Theorem 5 (2)]	ℓ_2/ℓ_1	N	$k \log^2 N$	$k \log^2 N$	$O\left(\log k \cdot \log\left(k \log N\right)\right) =^* O(\log^2 k)$
[Iwe14, Theorem 5 (3)]	ℓ_2/ℓ_1	N	$k \log k \cdot \log N$	$k \log k \cdot \log N$	$O\left(k \log k \cdot \log\left(k \log N\right)\right)$
Our Corollary 4.3.7	ℓ_2/ℓ_1	N	$k \log N$	$N \log N$	$O\left(\log N \cdot \log\left(k \log N\right)\right)$
Our Corollary 4.4.1	ℓ_2/ℓ_1	N	$k \log k \cdot \log N$	$k \log k \cdot \log N$	$O\left(\log k \cdot \log\left(k \log N\right)\right)$

Table 1.1: Summary of the principal previous results and the results obtained in this thesis. The column "D/U/N" specifies whether the error guarantee (1.1) applies to the deterministic (D) model, the uniform (U) recovery model or the nonuniform (N) recovery model. The measurement and runtime complexities are each subject to O-notation, suppressed for brevity. "LP" denotes the time complexity of solving a linear program of N variables. $*$ follows from the assumption of [Iwe14] that $k = \Omega(\log N)$. Some of the lower bounds reported in other papers have been refined here for comparison purposes. For example, [GLPS12, Theorem 1.1] reports the runtime of Δ in the less precise form $O(k \log^{O(1)} N)$. Schemes for which the entropy was not reported, but whose measurement matrix $\mathcal{M}$ contains at least one randomly generated entry in each column, are listed here as having entropy $\Omega(N)$.

1.3 Our Contributions

We now state our main contributions.

Fastest Deterministic ℓ_p/ℓ_q Compressed Sensing Scheme. In Corollary 3.5.1, we provide a deterministic compressed sensing scheme satisfying an ℓ_2/ℓ_1 error guarantee. Both the row count and the recovery algorithm runtime of this scheme are $O(k^2 \log^2 N)$. To the author's knowledge, the recovery algorithm is faster than all other deterministic compressed sensing schemes satisfying an ℓ_p/ℓ_q error guarantee. Furthermore, this scheme is memory-efficient. Our innovation lies in connecting ideas from different areas involving coding theory, group testing, and data streaming. In particular, we show how to use disjunct matrix constructions due to Porat and Rothschild [PR11] and to Kautz and Singleton [KS64] to produce certain measurement matrices. These matrices are paired with a sublinear-time recovery algorithm, inspired by a streaming algorithm, to produce the compressed sensing scheme.

Nonuniform ℓ_2/ℓ_1 Compressed Sensing Scheme with a Near-Optimal Runtime and Low Entropy. In Corollary 4.4.1, we provide a randomized variant of Corollary 3.5.1 that applies to the nonuniform recovery model. This compressed sensing scheme satisfies all four properties (P1)–(P4) simultaneously; to the author's knowledge, the only other scheme to do so is [Iwe14, Theorem 5 (2)]. Both the row count and the recovery algorithm runtime are $O(k \log k \cdot \log N)$. For the regime $k \leq N^c$ for some fixed $c \in [0, 1)$, these growth rates are within a factor of $O(\log k)$ of the respective lower bounds of $\Omega(k \log(N/k))$. In fact, our scheme is as fast as that of [Iwe14, Theorem 5 (3)], which has the fastest known runtime of schemes with an ℓ_p/ℓ_q error-guarantee, yet the entropy is reduced from $O\left(Nk^2 \log N\right)$ to $O\left(\log k \cdot \log\left(k \log N\right)\right)$. Furthermore, this entropy is as low as that of [Iwe14, Theorem 5 (2)], which is the only previously known scheme satisfying (P1)–(P3) and requiring sublinear entropy. In Corollary 4.3.7, we also provide a variant of Corollary 4.4.1, whose row-count m is optimal (for the regime $k \leq N^c$ for some fixed $c \in [0, 1)$). This is the first measurement-optimal scheme requiring sublinear entropy.

1.4 Thesis Outline

The thesis is organized as follows. Chapter 2 discusses background material on compressed sensing (Section 2.1) and related topics required to describe our main contributions (Sections 2.2, 2.3, and 2.4). In particular, Section 2.4 introduces (K, α)-coherent matrices, which will be used in constructing measurement matrices in later chapters.

Chapter 3 describes our first main contribution. The compressed sensing scheme is modularized into an identification scheme, estimation scheme and pruning algorithm (see Figure 3.1). We use a top-down approach and discuss these three modules in reverse order: pruning algorithm in Section 3.2, estimation scheme in Section 3.3, and identification scheme in Section 3.4. We combine these components in Section 3.5 to obtain our first main contribution.

Chapter 4 describes our second main contribution. This is obtained by randomizing the identification scheme and estimation scheme of Chapter 3. The randomized identification scheme and randomized estimation scheme are discussed in Sections 4.2 and 4.3, respectively. We combine them in Section 4.4 with the same pruning algorithm as in Section 3.2 to obtain our second main contribution (see Figure 4.1).

Chapter 5 contains concluding remarks and discussion of future work.

A preliminary version of some of our results was given in [LJ21] and presented at the online 2021 International Symposium on Information Theory.

Chapter 2

Background

2.1 Compressed Sensing

2.1.1 Geometric and Combinatorial Recovery Algorithms

Broadly speaking, the recovery algorithms of compressed sensing schemes follow either a geometric approach or a combinatorial approach.

Geometric approach. The geometric approach was first proposed in [CRT06a, Don06] and has been extensively studied since then. The key property of the measurement matrix which allows accurate recovery is the Restricted Isometry Property (see Definition 2.1.1). If a measurement matrix satisfies this property, then an approximation $\widehat{\mathbf{x}}$ can be obtained from the measurement $\mathbf{y} = \mathcal{M}\mathbf{x}$ by solving the following basis pursuit or ℓ_1-minimization program with N variables and m constraints, which is equivalent to solving a linear program with $2N$ variables and $m + 2N$ constraints [Til15]:

$$\text{minimise } \|\widehat{\mathbf{x}}\|_1 \text{ subject to } \mathcal{M}\widehat{\mathbf{x}} = \mathbf{y}. \tag{2.1}$$

The main advantage of the geometric approach is the small number of measurements: see Section 2.1.2 for some constructions. The main disadvantage of the geometric approach is the runtime of the recovery algorithm, since solving a linear program with $\Theta(N)$ variables and $\Omega(N)$ constraints takes $\Omega(N^\omega)$ time in general[1] where ω is the exponent of matrix multiplication.

[1] However, algorithms designed specifically for basis pursuit may be faster, see [FR13, Chapter 15] for some examples. Furthermore, the structure of $\mathcal{M}$ such as sparsity can also be exploited to have a faster runtime.

Combinatorial Approach. The combinatorial approach was first proposed in [CM06, Mut06, Ind08], using ideas related to group testing (see Section 2.3) and streaming algorithms [Mut06, GI10]. The recovery algorithm of the combinatorial approach is usually iterative, and each iteration comprises three phases: an identification phase, an estimation phase, and a pruning phase. In the identification phase, we identify a set $S \subseteq [N] = \{0, 1, \ldots, N-1\}$ containing the "heavy hitters" (the indices of the entries of $\mathbf{x}$ having largest magnitude), potentially with some false positives. In the estimation phase, we construct a vector $\mathbf{z} \in \mathbb{R}^N$ that estimates the entries of $\mathbf{x}$ at each index in S, and takes all other entries of $\mathbf{z}$ to be 0. In the pruning phase, we obtain an approximation $\widehat{\mathbf{x}}$ by setting to 0 all but $O(k)$ entries of $\mathbf{z}$ having the largest magnitude. See Algorithms 1 and 2 for an overview of the non-iterative and iterative versions, respectively, of the combinatorial approach. Our contributions are schemes using the non-iterative combinatorial approach.

The main advantage of the combinatorial approach is that it can be designed to have a fast recovery algorithm. Despite this, it is difficult to reduce the number of measurements to be as close to optimal as in the geometric approach.

Algorithm 1 Recovery Algorithm of Non-iterative Combinatorial Approach

```
Input: $\mathcal{M}$, $\mathbf{y}$, $k \in [N]$
Output: an approximation $\widehat{\mathbf{x}} \in \mathbb{R}^N$ to a best $k$-term approximation to $\mathbf{x}$
$S \leftarrow$ Identify($\mathcal{M}, \mathbf{y}$)        ▷ identify a set of indices containing the heavy hitters
$\mathbf{z} \leftarrow$ Estimate($\mathcal{M}, \mathbf{y}, S$)        ▷ estimate the value of $\mathbf{x}$ at each index in $S$
$\widehat{\mathbf{x}} \leftarrow$ Prune($\mathbf{z}, k, S$)        ▷ prune $\mathbf{z}$ to $O(k)$ nonzero entries
Output: $\widehat{\mathbf{x}}$
```

Algorithm 2 Recovery Algorithm of Iterative Combinatorial Approach [Li13, Algorithm 1.1]

```
Input: $\mathcal{M}$, $\mathbf{y}$, $k \in [N]$
Output: an approximation $\widehat{\mathbf{x}} \in \mathbb{R}^N$ to a best $k$-term approximation to $\mathbf{x}$
Initialize $\widehat{\mathbf{x}} \leftarrow \mathbf{0} \in \mathbb{R}^N$
while the halting criterion is not satisfied do
                ▷ at each iteration, update the partial approximation $\widehat{\mathbf{x}}$ to the final output
    $S \leftarrow$ Identify($\mathcal{M}, \mathbf{y}, \widehat{\mathbf{x}}$)
    $\mathbf{z} \leftarrow$ Estimate($\mathcal{M}, \mathbf{y}, S$)
    $\mathbf{w} \leftarrow$ Prune($\mathbf{z}, k, S$)
    $\widehat{\mathbf{x}} \leftarrow$ Merge($\widehat{\mathbf{x}}, \mathbf{w}$)        ▷ the partial approximation $\widehat{\mathbf{x}}$ converges
    $\mathbf{y} \leftarrow \mathbf{y} - \mathcal{M}\widehat{\mathbf{x}}$
end while
Output: $\widehat{\mathbf{x}}$
```

2.1.2 Restricted Isometry Property

In this section we introduce the Restricted Isometry Property of a measurement matrix. Although this property is primarily associated with geometric compressed sensing schemes, it is important in the development of ideas that led to our combinatorial compressed sensing schemes.

Definition 2.1.1 (Restricted Isometry Property (RIP))**.** Let $k \in \mathbb{N}$, and $\delta \in (0,1)$. An $m \times N$ matrix $\mathcal{M}$ has the Restricted Isometry Property of order k with constant δ (abbreviated as RIP(k,δ)) if

$$(1-\delta)\|\mathbf{x}\|_2^2 \leq \|\mathcal{M}\mathbf{x}\|_2^2 \leq (1+\delta)\|\mathbf{x}\|_2^2 \quad \text{for all } k\text{-sparse } \mathbf{x} \in \mathbb{R}^N. \tag{2.2}$$

For a matrix $\mathcal{M} \in \mathbb{C}^{m\times N}$ and positive integer k, it is desirable that (2.2) holds for small δ. For instance, if $\mathcal{M}$ has the RIP$(2k,\delta)$ with $\delta < \sqrt{2}-1$, then $\mathcal{M}$ can be combined with an ℓ_1-minimization-based recovery algorithm (2.1) to provide a *uniform* ℓ_2/ℓ_1 error guarantee [Can08]. Furthermore, RIP matrices for which (2.2) holds for sufficiently small δ can also be paired with other faster (although still $\Omega(N)$) recovery algorithms such as Regularized Orthogonal Matching Pursuit [NV09, NV10], CoSaMP [NT09], and Iterative Hard Thresholding [BD09]. Henceforth, when referring to RIP matrices, we implicitly mean those with small δ.

All known deterministic constructions of RIP matrices have poor row count. In fact, most of them [DeV07, HS09, AHSC09, Che11b, LGGZ12, CJ10, NT11, AM11, AMM12, YZ13, LG14a, LG14b, NJS16] are based on coherence [DE03], resulting in the restriction that $k = O(\sqrt{m})$ [Wel74, JMF13], commonly known as the square-root bottleneck. Consequently, the measurement matrix has $m = \Omega(k^2)$ rows and so does not satisfy property (P2) (a small row count). The only known deterministic RIP matrix construction that overcomes this bottleneck is given in [BDF$^+$11, Mix15], satisfying $k = \Omega(m^{1/2+\varepsilon_0})$ for the constant $\varepsilon_0 \approx 4.4466 \times 10^{-24}$ by [Mix15].

In contrast, randomized constructions of RIP matrices are not subject to the square-root bottleneck. For example, by sampling an $m \times N$ matrix with independent identically distributed Gaussian $\mathcal{N}(0,1/m)$ entries, one obtains an RIP(k,δ) matrix with probability $1-\epsilon$ provided that $m \geq C\delta^{-2}k\log(N/k)$, where C depends only on ϵ [CRT06a, Don06]. For fixed ϵ and δ, this measurement matrix has $m = O(k\log(N/k))$ rows and hence satisfies property (P2). Other notable randomized constructions of RIP matrices include Bernoulli random matrices [CT06a] and Fourier random matrices [CT06a, RV08, Bou14, NPW14, HR17, BLM18]. In fact, the row count for Gaussian and for Bernoulli random matrices is order-optimal [BIPW10, PW11]. However, these constructions require an entropy of $\Omega(N)$, and so do not satisfy property (P4) (low entropy). Furthermore, although these random matrices are very likely to satisfy the RIP, certifying this property is in general NP-hard [BDMS13, TP13, Wee17, WBP16, DKWB20]. Since this is true in particular for the regime $k \gg \sqrt{m}$, it is not practical to obtain an RIP matrix that overcomes the square-root bottleneck by presenting successive random matrices to a certifier until a "yes" output is obtained (so that the input matrix has been certified to have the RIP).

2.1.3 DeVore RIP Matrix Construction

Most deterministic RIP matrix constructions rely on systems of unit vectors (the columns of the matrix) having small coherence.

Definition 2.1.2 (μ-coherent)**.** Let $\mu \in [0, 1]$. An $m \times N$ matrix $\mathcal{M}$ is μ-coherent if each column is ℓ_2-normalized and the inner product between each distinct pair of columns is at most μ.

The following result shows that matrices with small coherence give good RIP properties.

Theorem 2.1.3 ([HR17, Theorem 5.3])**.** Suppose that an $m \times N$ matrix $\mathcal{M}$ is μ-coherent. Then $\mathcal{M}$ has the RIP$(k, (k-1)\mu)$ for all $k \in [N]$.

The first known deterministic construction of RIP matrices is due to DeVore [DeV07], and is obtained by applying Theorem 2.1.3 to the following result.

Theorem 2.1.4 (DeVore RIP Matrix [DeV07, Theorem 3.1])**.** Let q be a prime power, let k be a positive integer where $k \leq q$, and let $N = q^k$. Let $\mathbb{F}_q = \{\alpha_1, \ldots, \alpha_q\}$. Let P be the set of polynomials $\{f(y) = \sum_{i=0}^{k-1} b_i y^i \mid b_i \in \mathbb{F}_q\}$ of size N. Associate each $f \in P$ with a vector $\mathbf{x} = (x_0, \ldots, x_{q^2-1})$ of length q^2 by setting

$$x_{j+qi} = \begin{cases} \frac{1}{\sqrt{q}} & \text{if } f(\alpha_i) = j \\ 0 & \text{otherwise,} \end{cases}$$

where $0 \leq i \leq q-1$ and $0 \leq j \leq q-1$. (Each of these vectors is formed from q blocks of length q, where each block contains exactly one nonzero entry and this entry has value $\frac{1}{\sqrt{q}}$.) The $q^2 \times N$ matrix whose columns are given by these N vectors is $\frac{\log_q N - 1}{q}$-coherent.

In Section 2.4.2, we will connect the DeVore RIP Matrix to Reed-Solomon codes (see Section 2.2.2), combinatorial group testing (see Section 2.3), and incoherent binary matrices (see Section 2.4).

2.2 Coding Theory

The DeVore RIP Matrix is related to Reed-Solomon codes, even though this is not explicitly stated in [DeV07]. To the author's knowledge, this connection was first explicitly stated in [MB11, Section III. B.]. In fact, the measurement matrix of many compressed sensing schemes [CM06, HCS08, AHSC09, MB11, DSV12, LGGZ12, AMM12, YZ13, LV20] can be viewed as a construction from codes.

2.2.1 Basic Concepts

Definition 2.2.1 (code, binary code, codeword, length, size)**.** A code C over a finite field $\mathbb{F}_q$ is a subset of $(\mathbb{F}_q)^m$, where the positive integer m is the length and $|C|$ is the size of the code. Each member of C is called a codeword. The code is binary if $q = 2$.

Definition 2.2.2 (distance)**.** The distance $d(c, c')$ between codewords c and c' is the number of positions in which c and c' differ. The distance d of a code C is the minimum of the distances between each two distinct codewords of C.

Definition 2.2.3 (weight, constant weight)**.** The weight of a codeword c is the number of non-zero entries in c. A code of constant weight w is one for which all codewords have equal weight w. We denote a length m code over $\mathbb{F}_q$ of distance d and constant weight w as an $(m, d, w)_q$-code.

Definition 2.2.4 (linear code, dimension)**.** A length m code C over $\mathbb{F}_q$ is linear if it is a subspace of $(\mathbb{F}_q)^m$ (regarded as a vector space under vector addition). The dimension k of a length m linear code C over $\mathbb{F}_q$ is the dimension of the subspace C (so that $k \leq m$ and $|C| = q^k$). We denote a linear code over $\mathbb{F}_q$ of length m, dimension k, and distance d, as an $[m, k, d]_q$-code.

2.2.2 Reed-Solomon Codes

One of the most important problems in coding theory is: for fixed m, d, and q, what is the maximum size N of a length m code of distance d over $\mathbb{F}_q$? The Singleton bound [Sin64] gives a simple upper bound on this size. This gives a necessary condition for the existence of a code with respect to the parameters N, m, d, q.

Theorem 2.2.5 (Singleton bound [Rot06, Theorem 4.1], MDS code)**.** Suppose that C is a code over $\mathbb{F}_q$, with length m, size N and distance d. Then $d \leq m - \log_q N + 1$. A maximum distance separable (MDS) code is one that meets the Singleton bound.

We now describe a special class of codes meeting the Singleton bound, namely the Reed-Solomon codes [RS60].

Definition 2.2.6 (Reed-Solomon code)**.** Let $k \leq r \leq q$ be positive integers where q is a prime power. Let $\alpha_1, \cdots, \alpha_r$ be distinct elements of $\mathbb{F}_q$. Consider the set of polynomials $\{f(x) = \sum_{i=0}^{k-1} b_i x^i \mid b_i \in \mathbb{F}_q\}$ of size q^k. Associate each polynomial f with the length r word over $\mathbb{F}_q$ given by $(f(\alpha_1), \cdots, f(\alpha_r))$. The set of all such words is the Reed-Solomon code, which is a linear $[r, k, r-k+1]_q$-code and so is an MDS code.

2.2.3 Porat-Rothschild Construction of Linear Codes Meeting the Gilbert-Varshamov Bound

Whereas the Singleton bound provides a necessary condition for the existence of a code with given parameters, we now present a sufficient condition. This was proved for general codes by Gilbert [Gil52] and for linear codes by Varshamov [Var57]. We will describe the case for linear codes below, which requires the following definition.

Definition 2.2.7 (q-ary entropy)**.** Let $q \geq 2$ be an integer. The q-ary entropy function $H_q \colon [0,1] \to \mathbb{R}$ is defined by

$$H_q(\delta) = \begin{cases} 0 & \text{for } \delta \in \{0,1\} \\ \delta \log_q(q-1) - \delta \log_q(\delta) - (1-\delta)\log_q(1-\delta) & \text{otherwise.} \end{cases}$$

Theorem 2.2.8 (Asymptotic Gilbert–Varshamov (GV) bound for linear codes [Rot06, Theorem 4.10])**.** Let $q \geq 2$ be a prime power, and let $0 \leq \delta \leq 1 - 1/q$. Let $m \geq 2$ be an integer, and let k be an integer such that $k \leq (1 - H_q(\delta))m$. Then there exists an $[m, k, \delta m]_q$-code.

The original proof for Theorem 2.2.8 uses a standard probabilistic method that is non-constructive. Using a derandomization method known as the method of conditional expectation [AS16], Porat and Rothschild [PR11] gave an explicit construction of such codes, which we now summarize.

Theorem 2.2.9 (Explicit linear code meeting GV bound [PR11, Theorem 3])**.** Let $q \geq 2$ be a prime power, and let $0 < \delta \leq 1 - 1/q$. Let $m \geq 2$ be an integer, and let k be an integer such that $k \leq (1 - H_q(\delta))m$. Then a $[m, k, \delta m]_q$-code can be explicitly constructed in $\Theta(q^k m)$ time.

2.2.4 Kautz-Singleton Concatenation

Code concatenation [For65] is a technique for constructing a long code over a small field with good properties from a long code over a large field and a short code over a small field. A particular example, known as Kautz-Singleton concatenation [KS64], can be used to construct binary codes with good properties from codes over $\mathbb{F}_q$. This technique will be used in Sections 2.3 and 2.4 to construct disjunct matrices and incoherent binary matrices. For more discussion about general code concatenation, see [GRS19, §10].

Definition 2.2.10 (Kautz-Singleton concatenation)**.** Let α be a primitive element of $\mathbb{F}_q$, so that $\mathbb{F}_q = \{0, 1, \alpha, \ldots, \alpha^{q-3}, \alpha^{q-2}\}$. Suppose C is a code over $\mathbb{F}_q$ with length m and distance d. Then the image of C under the mapping that replaces symbols of $\mathbb{F}_q$ by length q binary vectors according to the rule

$$\begin{aligned}
0 &\mapsto 100\ldots00 \\
1 &\mapsto 010\ldots00 \\
\alpha &\mapsto 001\ldots00 \\
&\vdots \\
\alpha^{q-3} &\mapsto 000\ldots10 \\
\alpha^{q-2} &\mapsto 000\ldots01
\end{aligned}$$

is a binary constant-weight $(mq, 2d, m)_2$-code.

2.3 Group Testing

Non-adaptive group testing [KS64, Kat73, DR82, DH00, DH06, AJS19] can be thought of as a compressed sensing problem over a Boolean domain $\{0, 1\}$. In this setting, the $m \times N$ measurement matrix $\mathcal{M} = (m_{i,j})$ and the unknown k-sparse vector $\mathbf{x} = (x_j)$ are both binary (having entries in $\{0, 1\}$), and the addition and multiplication use Boolean algebra. We write $\vee$ for the Boolean sum (disjunction), and $\wedge$ for the Boolean product (conjunction). The measurement $\mathbf{y} = (y_i)$ is defined by the logical expression

$$y_i \coloneqq (m_{i,1} \wedge x_1) \vee (m_{i,2} \wedge x_2) \vee \cdots \vee (m_{i,N} \wedge x_N).$$

Since $y_i = m_{i,1}x_1 \vee \cdots \vee m_{i,N}x_N$, the distinction between this setting and the previous setting $y_i = m_{i,1}x_1 + \cdots + m_{i,N}x_N$ is that addition over $\mathbb{R}$ is replaced by the Boolean sum. As before, we wish to approximate $\mathbf{x}_k$ from $\mathcal{M}$ and $\mathbf{y}$, and we want the row count m of $\mathcal{M}$ to be small. In this case $\mathbf{x}_k = \mathbf{x}$ since $\mathbf{x}$ is k-sparse.

The non-adaptive group testing problem has recently attracted particular attention in the context of COVID-19 testing [Täu20, VFH+20, Nal20, MSW+21, Seo20, ZDL+21]. In this context, N is the number of individuals to be tested, k is the number of infected people, and m is the number of group tests. The unknown vector $\mathbf{x} = (x_j) \in \{0,1\}^N$ represents the infection status of the N individuals, where $x_j = 1$ if and only if individual j is infected. We wish to determine the indices j for which $x_j = 1$ (representing the infected individuals) by carrying out m group tests, each of which pools biological sample material from a subset of the individuals. The pooling design is represented by the measurement matrix $\mathcal{M}$: group test i contains sample material from individual j if and only if $m_{i,j} = 1$. The outcome of group test i is represented by y_i, which takes the value 1 if and only if group test i contains sample material from an infected individual (that is, there exists some j for which $x_j = 1$ and $m_{i,j} = 1$).

2.3.1 Disjunct Matrices

In combinatorial group testing, we wish to recover the unknown k-sparse vector $\mathbf{x}$ exactly. The following combinatorial structure for the binary matrix $\mathcal{M}$ allows us to do so.

Definition 2.3.1 (k-disjunct)**.** A binary vector (u_i) contains a binary vector (v_i) of equal length if, for each i,

$$v_i = 1 \implies u_i = 1.$$

A binary matrix $\mathcal{M}$ is k-disjunct[2] if there is no set S of k columns of $\mathcal{M}$ whose Boolean sum contains a column of $\mathcal{M}$ not in S.

Provided the binary matrix $\mathcal{M}$ is k-disjunct, the k-sparse vector $\mathbf{x}$ can be recovered from $\mathbf{y}$ using the naive $O(mN)$-time[3] recovery algorithm:

$$x_j = 0 \text{ if and only if there exists some } i \text{ for which } y_i = 0 \text{ and } m_{i,j} = 1.$$

In the COVID-19 testing context, we certify each individual appearing in a negative group test as healthy, and all other individuals as infected. We remark the underlying idea of this group testing recovery algorithm has appeared under many names in [KS64, Mal13, CCJS11, CJSA14, LG08]. It can be viewed as a combinatorial analogue of the Orthogonal Matching

[2] also known as zero-false-drop (*ZFD*) code [KS64], disjunctive code [DR82], cover-free family [EFF85, Rus94, Fü96], and strongly selective family [CMS03])

[3] This algorithm can be paired with an arbitrary k-disjunct matrix, but its runtime is impractical for large N. In contrast, there are specific k-disjunct matrices which can be paired with their associated sublinear-time recovery algorithm [INR10, NPR11, CN20].

Pursuit recovery algorithm in compressed sensing [TG07], and so is now commonly known as Combinatorial Orthogonal Matching Pursuit [CCJS11]. See [GIS08] for more detailed exposition on the extensive interplay between compressed sensing and group testing.

We now give a lower bound on the growth rate of the row count m of a k-disjunct matrix of N columns (in which we consider N and k to be varying parameters).

Theorem 2.3.2 (lower bound for disjunct matrix [DR82])**.** Let N and k be positive integers such that $k < N$. Then a k-disjunct matrix of size $m \times N$ has $m = \Omega\Big(\min\left(k^2 \log_k N, N\right)\Big)$.

The best known probabilistic upper bound and constructive upper bound are both very close to the lower bound of Theorem 2.3.2.

Theorem 2.3.3 (probabilistic upper bound for disjunct matrix [Che11a, Theorem 4.4])**.** Let N and k be positive integers such that $k < N$. Then there exists a k-disjunct matrix of size $m \times N$ with $m = O\Big(\min\left(k^2 \log(N/k), N\right)\Big)$.

The best row count for an explicit construction of a k-disjunct matrix to date is due to Porat and Rothschild [PR11]. They apply Kautz-Singleton concatenation (see Definition 2.2.10) to the explicit linear code meeting the Gilbert-Varshamov bound (produced in the same paper, see Theorem 2.2.9) to obtain a binary constant-weight code. The associated binary matrix (whose columns are the codewords) is a k-disjunct matrix whose row count is bounded as follows.

Theorem 2.3.4 (constructive upper bound for disjunct matrix [PR11])**.** Let N and k be positive integers such that $k < N$. Then a k-disjunct matrix of size $m \times N$ with $m = O\Big(\min\left(k^2 \log N, N\right)\Big)$ can be explicitly constructed in $\Theta(NK)$ time.

2.4 Binary Matrices with Small Coherence

In Section 2.1.3 we defined μ-coherent matrices and described a connection with RIP matrices. We now introduce a modified definition of coherence of binary matrices, to be used in constructing a compressed sensing measurement matrix $\mathcal{M}$ in Chapter 3. We will discuss the relationships between binary matrices with small coherence and RIP matrices, codes, and disjunct matrices (see Figure 2.1). See also [Che11b] for an overview of these relationships, as well as those involving codes with large distance, spherical codes with small coherence, list-decodable codes, and combinatorial designs.

2.4.1 (K, α)-Coherent Matrices

Definition 2.4.1 ((K, α)-coherent matrix)**.** Let K and α be positive integers satisfying $K > \alpha$. An $m \times N$ binary $\{0, 1\}$ matrix $\mathcal{M}_{\mathrm{C}}$ is $\underline{(K, \alpha)\text{-coherent}}$ if each column of $\mathcal{M}_{\mathrm{C}}$ contains exactly K ones and each pair of distinct columns of $\mathcal{M}_{\mathrm{C}}$ has dot product at most α.

Remark 2.4.2 ((K, α)-coherent matrix is $\frac{\alpha}{K}$-coherent)**.** If we apply ℓ_2 normalization to each column of a (K, α)-coherent matrix $\mathcal{M}_{\mathrm{C}}$, then the resulting matrix is $\frac{\alpha}{K}$-coherent (see Definition 2.1.2) and therefore has the RIP$(k, (k-1)\frac{\alpha}{K})$ for all $k \in [N]$ (see Theorem 2.1.3).

Remark 2.4.3. By viewing the codewords of a binary constant-weight $(m, 2(K-\alpha), K)_2$-code of size N (see Definition 2.2.3) as columns of a binary matrix, we obtain a (K, α)-coherent matrix of size $m \times N$ for which α is tight (there exists a pair of columns whose dot product is exactly α).

In [Joh62, Theorem 3], Johnson introduced an auxiliary function associated with the well-known problem of determining the maximum size of a length m binary code of distance d and constant weight w, and derived an upper bound on this function using elementary methods. We now obtain the following lower bound on the row count of a (K, α)-coherent matrix by making a connection with this auxiliary function.

Theorem 2.4.4. Let N, K, and α be positive integers such that $K > \alpha$. Let $\mathcal{M}_{\mathrm{C}} \in \{0, 1\}^{m \times N}$ be a (K, α)-coherent matrix. Then

$$m \geq \frac{NK^2}{(N-1)\alpha + K} = \Omega\left(\min\left(\frac{K^2}{\alpha}, \frac{NK}{\alpha}\right)\right).$$

We now show that a (K, α)-coherent matrix $\mathcal{M}_{\mathrm{C}}$ is also a disjunct matrix (see Definition 2.3.1).

Theorem 2.4.5 ([DH00, Lemma 7.3.2])**.** A (K, α)-coherent matrix $\mathcal{M}_{\mathrm{C}}$ of size $m \times N$ is $\lfloor (K-1)/\alpha \rfloor$-disjunct.

Proof. Let S be a set of $\lfloor (K-1)/\alpha \rfloor$ columns of $\mathcal{M}_{\mathrm{C}}$ and let $\mathbf{v}$ be a column of $\mathcal{M}_{\mathrm{C}}$ not in S. By the definition of a (K, α)-coherent matrix, each vector in S has at most α entries 1 in common with $\mathbf{v}$. Therefore the Boolean sum of the vectors in S has at most $\alpha \lfloor (K-1)/\alpha \rfloor \leq K - 1$ entries 1 in common with $\mathbf{v}$ and so does not contain $\mathbf{v}$. □

We summarize the relationships between these combinatorial objects in Figure 2.1.

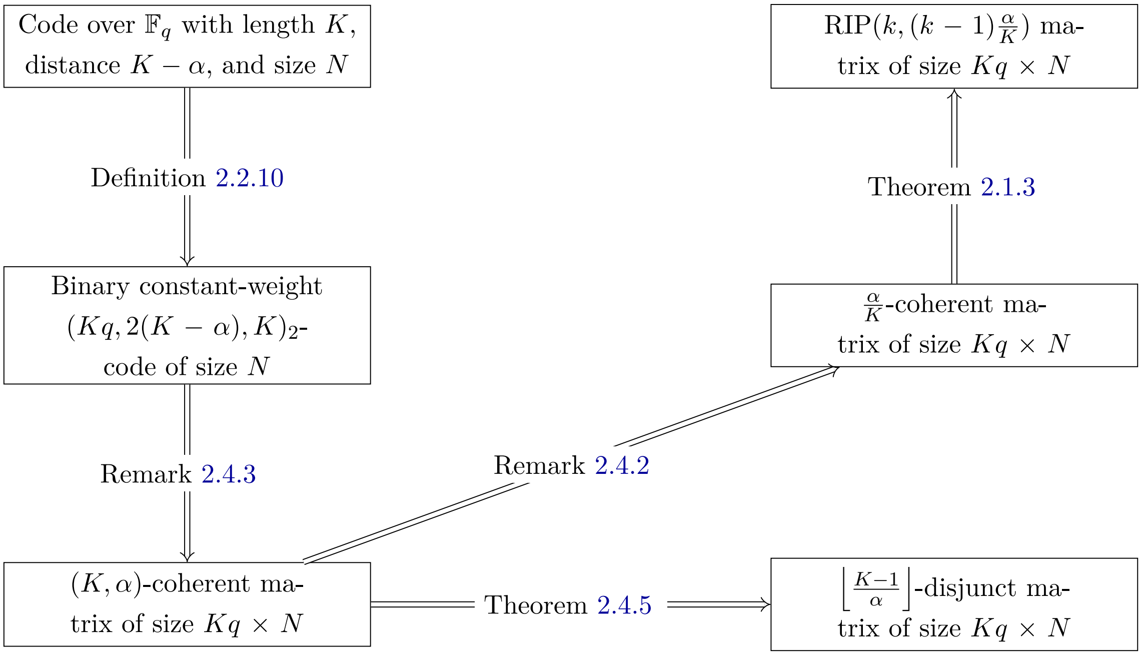

Code over $\mathbb{F}_q$ with length K, distance $K-\alpha$, and size N
Definition 2.2.10
Binary constant-weight $(Kq, 2(K-\alpha), K)_2$-code of size N
Remark 2.4.3
(K,α)-coherent matrix of size $Kq \times N$
Remark 2.4.2
Theorem 2.4.5
$\left\lfloor \frac{K-1}{\alpha} \right\rfloor$-disjunct matrix of size $Kq \times N$
$\frac{\alpha}{K}$-coherent matrix of size $Kq \times N$
Theorem 2.1.3
RIP$(k, (k-1)\frac{\alpha}{K})$ matrix of size $Kq \times N$

Figure 2.1: Relationships between code with large distance, (K, α)-coherent matrix, disjunct matrix, and RIP matrix.

2.4.2 Reinterpretation of DeVore RIP Matrix as Kautz-Singleton Concatenation of Reed-Solomon Code

Kautz and Singleton [KS64, Section V B] gave a strongly explicit construction of a disjunct matrix[4], using Kautz-Singleton concatenation to transform a Reed-Solomon code [RS60] into a binary constant-weight code and then taking the associated matrix. After normalizing the columns, we recognize the resulting matrix as being identical to the DeVore RIP matrix (see Sections 2.1.3 and 2.2). Furthermore, this disjunct matrix is also (K, α)-coherent for a sufficiently large value of α (even though the converse of Theorem 2.4.5 does not hold in general). The following theorem describes this reinterpretation of the DeVore RIP matrix construction. It is essentially Theorem 2.1.4, in which each nonzero entry $\frac{1}{\sqrt{q}}$ is replaced by 1, and q is chosen to be the first prime power at least as large as K (so that $K \le q < 2K$ by Bertrand's postulate).

Theorem 2.4.6. Let N, K, α be positive integers satisfying $K > \alpha \ge \log_K N - 1$. Then we can explicitly construct a (K, α)-coherent matrix $\mathcal{M}_{\text{KS}}$ of size $m \times N$, where $m = \Theta(K^2)$. Furthermore, the matrix comprises exactly K blocks of $\Theta(K)$ rows, each column of each row block containing exactly one 1. Furthermore, the location of the unique 1 entry in row

[4] A construction of an $m \times N$ matrix is strongly explicit if each column of the matrix can be constructed in time poly(m), and explicit if each column can be constructed in time poly(m, N).

block $j \in [K]$ and column $n \in [N]$ can be determined from j and n in $O(\log_K N)$ time using Horner's rule

$$a_n x_n + a_n x_{n-1} + \cdots + a_0 = \left(\left(a_n x + a_{n-1}\right) x + \cdots\right) x + a_o.$$

We summarize the relationships between these combinatorial objects in Figure 2.2.

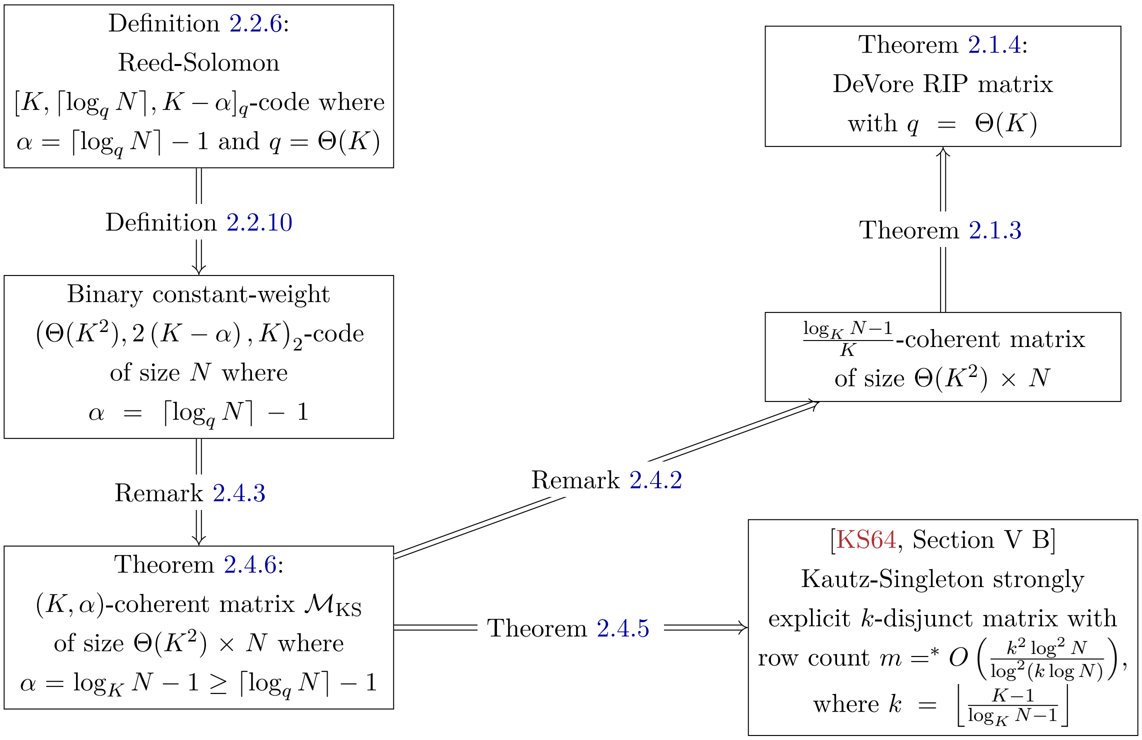

Definition 2.2.6:
Reed-Solomon
$[K, \lceil \log_q N \rceil, K-\alpha]_q$-code where
$\alpha = \lceil \log_q N \rceil - 1$ and $q = \Theta(K)$
Definition 2.2.10
Binary constant-weight
$(\Theta(K^2), 2(K-\alpha), K)_2$-code
of size N where
$\alpha = \lceil \log_q N \rceil - 1$
Remark 2.4.3
Theorem 2.4.6:
(K, α)-coherent matrix $\mathcal{M}_{\text{KS}}$
of size $\Theta(K^2) \times N$ where
$\alpha = \log_K N - 1 \geq \lceil \log_q N \rceil - 1$
Theorem 2.1.4:
DeVore RIP matrix
with $q = \Theta(K)$
Theorem 2.1.3
$\frac{\log_K N - 1}{K}$-coherent matrix
of size $\Theta(K^2) \times N$
Remark 2.4.2
Theorem 2.4.5
[KS64, Section V B]
Kautz-Singleton strongly
explicit k-disjunct matrix with
row count $m =^* O\left(\frac{k^2 \log^2 N}{\log^2(k \log N)}\right)$,
where $k = \left\lfloor \frac{K-1}{\log_K N - 1} \right\rfloor$

Figure 2.2: Relationships between Reed-Solomon code, $(K, \log_K N - 1)$-coherent matrix, Kautz-Singleton disjunct matrix, and DeVore RIP matrix. All matrix constructions are strongly explicit. * follows from $K^2 = O\left(\frac{k^2 \log^2 N}{\log^2 K}\right) = O\left(\frac{k^2 \log^2 N}{\log^2(K \log K)}\right) = O\left(\frac{k^2 \log^2 N}{\log^2(k \log N)}\right)$.

2.4.3 Order-Optimal Construction: Reinterpretation of Porat-Rothschild Disjunct Matrix Construction

In Section 2.4.2 we made a connection between a Reed-Solomon code, a Kautz-Singleton disjunct matrix, and the DeVore RIP matrix, reinterpreting them as a (K, α)-coherent matrix (see Figure 2.2). We now make an analogous connection between a code meeting the Gilbert-Varshamov bound (Theorem 2.2.9), a Porat-Rothschild disjunct matrix (Theorem 2.3.4), and an RIP matrix introduced by Cheraghchi [Che11b, Section III], reinterpreting them as a (K, α)-coherent matrix (see Figure 2.3).

The following theorem describes this reinterpretation of the Porat-Rothschild disjunct matrix as a $(\Theta(K), \Theta(\log N))$-coherent matrix, whose row count is order-optimal for $N > K$ by Theorem 2.4.4.

Theorem 2.4.7. Let N, K, α be positive integers satisfying $K > \alpha > c \log N$ for some absolute constant $c > 0$. Then we can explicitly construct a (K, α)-coherent matrix $\mathcal{M}_{\mathrm{PR}}$ of size $m \times N$ in $\Theta(NK)$ time, where $m = \Theta(K^2/\alpha)$. Furthermore, the matrix comprises exactly K blocks of $\Theta(K/\alpha)$ rows, each column of each row block containing exactly one 1.

We summarize these relationships, as well as those involving the Porat-Rothschild construction of linear codes meeting the Gilbert-Varshamov bound, in Figure 2.3.

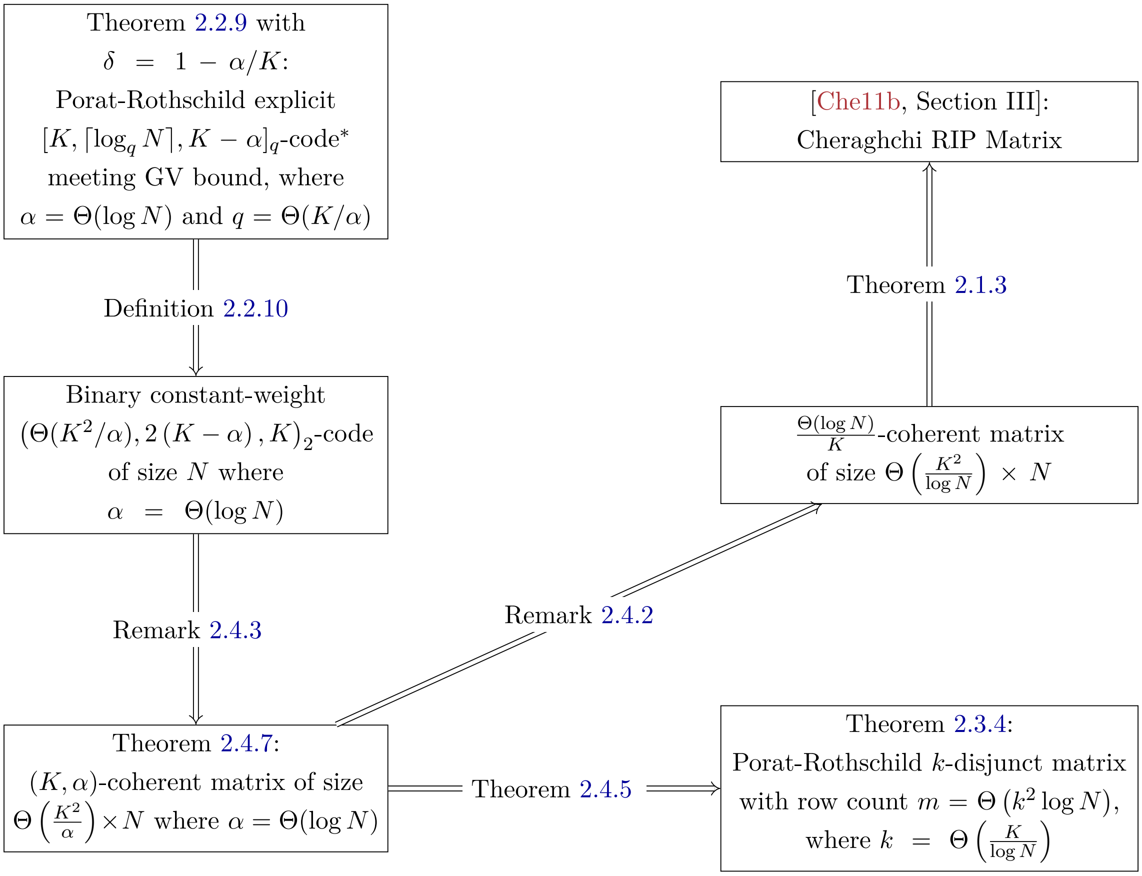

Theorem 2.2.9 with $\delta = 1 - \alpha/K$: Porat-Rothschild explicit $[K, \lceil \log_q N \rceil, K - \alpha]_q$-code* meeting GV bound, where $\alpha = \Theta(\log N)$ and $q = \Theta(K/\alpha)$
Definition 2.2.10
Binary constant-weight $(\Theta(K^2/\alpha), 2(K - \alpha), K)_2$-code of size N where $\alpha = \Theta(\log N)$
Remark 2.4.3
Theorem 2.4.7: (K, α)-coherent matrix of size $\Theta\left(\frac{K^2}{\alpha}\right) \times N$ where $\alpha = \Theta(\log N)$
Theorem 2.4.5
Theorem 2.3.4: Porat-Rothschild k-disjunct matrix with row count $m = \Theta(k^2 \log N)$, where $k = \Theta\left(\frac{K}{\log N}\right)$
Remark 2.4.2
$\frac{\Theta(\log N)}{K}$-coherent matrix of size $\Theta\left(\frac{K^2}{\log N}\right) \times N$
Theorem 2.1.3
[Che11b, Section III]: Cheraghchi RIP Matrix

Figure 2.3: Relationship between Porat-Rothschild explicit linear codes meeting GV bound, $(K, \Theta(\log N))$-coherent matrix, Porat-Rothschild explicit disjunct matrix, and Cheraghchi RIP matrix. All matrix constructions are explicit. The existence of a code* follows from $1 - H_q(1 - \frac{\alpha}{K}) = \frac{\log\left(1+\frac{1}{q-1}\right)}{\log q} + \frac{\alpha}{K}\frac{\log\left(\frac{(q-1)\alpha}{K}\right)}{\log q} + (1 - \frac{\alpha}{K})\frac{\log\left(1-\frac{\alpha}{K}\right)}{\log q} = \frac{1}{\log q}\left(\Theta\left(\frac{1}{q}\right) + \frac{\alpha}{K}\Theta(1) - \Theta\left(\frac{\alpha}{K}\right)\right) = \Theta\left(\frac{\alpha}{K \log q}\right)$ using $\log\left(1 \pm \frac{1}{x}\right) = \Theta\left(\pm\frac{1}{x}\right)$ and $q - 1 = \Theta(K/\alpha)$; therefore $\left(1 - H_q\left(1 - \frac{\alpha}{K}\right)\right) K = \Theta\left(\frac{\alpha}{\log q}\right) = \Theta(\log_q N)$.

Chapter 3

Deterministic Compressed Sensing Scheme with a Fast Runtime

In this chapter, we give a deterministic compressed sensing scheme satisfying an ℓ_2/ℓ_1 error guarantee (Corollary 3.5.1) which, to the author's knowledge, achieves a faster runtime than all other deterministic compressed sensing schemes satisfying an ℓ_p/ℓ_q error guarantee.

3.1 Overview of Techniques

Measurement matrix. The measurement matrix $\mathcal{M}$ takes the form $\mathcal{M} = \left[\begin{array}{c} \mathcal{M}_{\text{id}} \\ \hline \mathcal{M}_{\text{est}} \end{array}\right]$, where $\mathcal{M}_{\text{id}}$ is called an identification matrix and $\mathcal{M}_{\text{est}}$ is called an estimation matrix. The measurement $\mathbf{y} = \mathcal{M}\mathbf{x}$ can therefore be considered as a two-part measurement

$$\mathbf{y} = \left[\begin{array}{c} \mathcal{M}_{\text{id}} \\ \hline \mathcal{M}_{\text{est}} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{y}_{\text{id}} \\ \hline \mathbf{y}_{\text{est}} \end{array}\right].$$

See Figure 3.1 for an overview of how the identification measurement $\mathbf{y}_{\text{id}}$ and estimation measurement $\mathbf{y}_{\text{est}}$ are used in the recovery algorithm.

Recovery algorithm. Our main compressed sensing scheme in this chapter (Corollary 3.5.1) is a scheme using the non-iterative combinatorial approach, and the recovery algorithm Δ has the structure of Algorithm 1. Our recovery algorithm is the same as that of Iwen [BIS12, Algorithm 1], which we present here as Algorithm 8. The three phases (identification, estimation and pruning) of this algorithm are modularized and presented as Algorithms 7, 6 and 5 respectively. On a high level, this can be represented as Algorithm 3.

Algorithm 3 Summary of Iwen's Recovery Algorithm [BIS12, Algorithm 1]

```
Input: $\mathcal{M} = \left[\begin{array}{c} \mathcal{M}_{\text{id}} \\ \hline \mathcal{M}_{\text{est}} \end{array}\right]$, $\mathbf{y} = \left[\begin{array}{c} \mathbf{y}_{\text{id}} \\ \hline \mathbf{y}_{\text{est}} \end{array}\right] = \mathcal{M}\mathbf{x}$, and $k$
Output: an approximation $\widehat{\mathbf{x}} \in \mathbb{R}^N$ satisfying an $\ell_2/\ell_1$ error guarantee
$S =$ Algorithm 7$(\mathcal{M}_{\text{id}}, \mathbf{y}_{\text{id}})$        ▷ set of indices containing largest entries of $\mathbf{x}$
$\mathbf{z} =$ Algorithm 6$(\mathcal{M}_{\text{est}}, \mathbf{y}_{\text{est}}, S)$        ▷ estimate the value of $\mathbf{x}$ at each index in $S$
$\widehat{\mathbf{x}} =$ Algorithm 5$(\mathbf{z}, k, S)$        ▷ prune $\mathbf{z}$ to at most $2k$ nonzero entries
Output: $\widehat{\mathbf{x}} \in \mathbb{R}^N$
```

Each of these three phases, together with its corresponding matrix (if applicable), is discussed later: identification matrix $\mathcal{M}_{\text{id}}$ and Algorithm 7 in Section 3.4; estimation matrix $\mathcal{M}_{\text{est}}$ and Algorithm 6 in Section 3.3; and Algorithm 5 in Section 3.2.

Identification-free scheme. We also present a modification of the above scheme in which the measurement matrix consists of only the estimation matrix, and the recovery algorithm (see Algorithm 4) consists of only the estimation and pruning phase. That is, the identification of indices of the largest entries of $\mathbf{x}$ in Step 1 of Algorithm 3 is not carried out. Instead, we take $S = [N]$ as an input for Algorithm 6. Consequently, fewer measurements are needed but the recovery algorithm is slower as the entries are now estimated at each index in $[N]$.

Algorithm 4 Identification-free Variant of Algorithm 3

```
Input: $\mathcal{M} = \mathcal{M}_{\text{est}}$, $\mathbf{y} = \mathbf{y}_{\text{est}} = \mathcal{M}\mathbf{x}$, and $k$
Output: an approximation $\widehat{\mathbf{x}} \in \mathbb{R}^N$ satisfying an $\ell_2/\ell_1$ error guarantee (1.1)
$\mathbf{z} =$ Algorithm 6$(\mathcal{M}_{\text{est}}, \mathbf{y}_{\text{est}}, [N])$        ▷ estimate the value of $\mathbf{x}$ at each index in $[N]$
$\widehat{\mathbf{x}} =$ Algorithm 5$(\mathbf{z}, k, [N])$        ▷ prune $\mathbf{z}$ to at most $2k$ nonzero entries
Output: $\widehat{\mathbf{x}} \in \mathbb{R}^N$
```

Ingredients of the measurement matrix. Each of $\mathcal{M}_{\text{id}}$ and $\mathcal{M}_{\text{est}}$ is constructed from a (K, α)-coherent matrix (see Section 2.4): either a Porat-Rothschild matrix $\mathcal{M}_{\text{PR}}$ (having order-optimal row count), or a Kautz-Singleton matrix $\mathcal{M}_{\text{KS}}$.

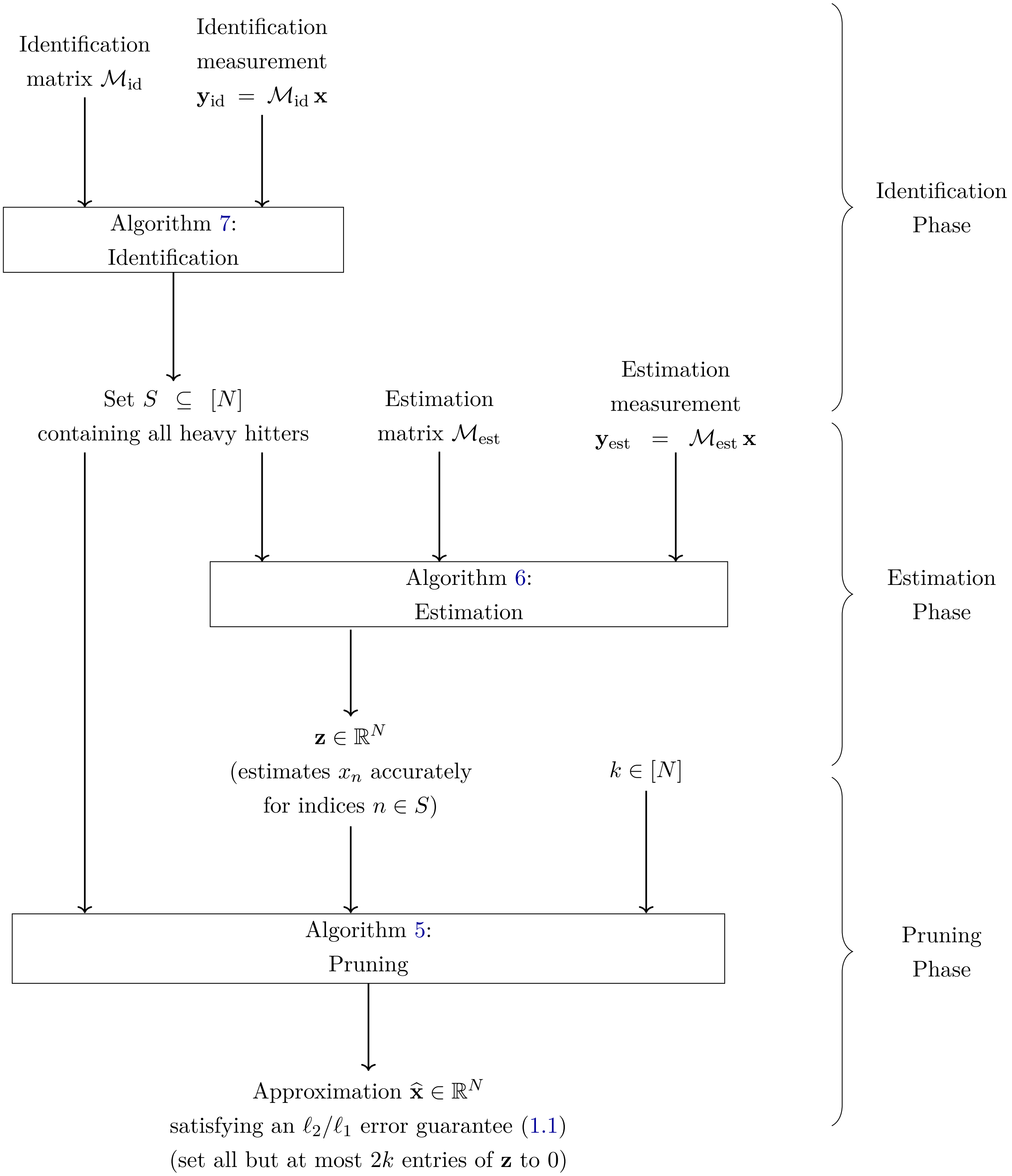

Identification matrix $\mathcal{M}_{\text{id}}$
Identification measurement $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\,\mathbf{x}$
Algorithm 7: Identification
Set $S \subseteq [N]$ containing all heavy hitters
Estimation matrix $\mathcal{M}_{\text{est}}$
Estimation measurement $\mathbf{y}_{\text{est}} = \mathcal{M}_{\text{est}}\,\mathbf{x}$
Algorithm 6: Estimation
$\mathbf{z} \in \mathbb{R}^N$ (estimates x_n accurately for indices $n \in S$)
$k \in [N]$
Algorithm 5: Pruning
Approximation $\widehat{\mathbf{x}} \in \mathbb{R}^N$ satisfying an ℓ_2/ℓ_1 error guarantee (1.1) (set all but at most $2k$ entries of $\mathbf{z}$ to 0)
Identification Phase
Estimation Phase
Pruning Phase

Figure 3.1: Overview of the identification, estimation, and pruning phases of the recovery algorithm (Algorithm 8) for the deterministic compressed sensing scheme of Corollary 3.5.1.

3.2 Pruning Algorithm

In this section, we describe the pruning phase depicted in Figure 3.1. The pruning algorithm takes as an input the vector $\mathbf{z} \in \mathbb{R}^N$ output by the estimation algorithm. This vector accurately estimates the entries of $\mathbf{x} \in \mathbb{R}^N$ at the indices identified by the identification algorithm, and is 0 at all other indices. In this section, we show that setting to 0 all but (at most) $2k$ entries of $\mathbf{z}$ having the largest magnitude gives an approximation to $\mathbf{x}$ satisfying an ℓ_2/ℓ_1 error guarantee. We formalize this in Theorem 3.2.3, and then generalize it to Corollary 3.2.4.

We first establish some notation and a lemma. We represent the entries of an arbitrary vector $\mathbf{x} \in \mathbb{R}^N$ as (x_i). Given $\mathbf{x} \in \mathbb{R}^N$ and a subset $S \subseteq [N]$, we define the restriction of $\mathbf{x}$ to S, denoted $\mathbf{x}_S \in \mathbb{R}^N$, by

$$(x_S)_i = \begin{cases} x_i & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Algorithm 5 Pruning (also known as hard thresholding operator with order $2k$)

```
Input: $\mathbf{z} \in \mathbb{R}^N$, $k \in [N]$, and $S \subseteq [N]$, where $\mathbf{z}_S = \mathbf{z}$
Output: $\widehat{\mathbf{x}} \in \mathbb{R}^N$
if $2k < |S|$ then
    Sort by magnitude the entries of $\mathbf{z}_S$ so that $|z_{n_1}| \geq |z_{n_2}| \geq \cdots \geq \left|z_{n_{|S|}}\right|$
    $\widetilde{S} \leftarrow \{n_1, \ldots, n_{2k}\}$
else
    $\widetilde{S} \leftarrow S$
end if
Output: $\widehat{\mathbf{x}} = \mathbf{z}_{\widetilde{S}}$                    ▷ $\left|\widetilde{S}\right| = \min(2k, |S|)$
```

Lemma 3.2.1. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. If

$$|x_n| > \frac{1}{k}\,\sigma_k(\mathbf{x})_1, \tag{3.1}$$

then x_n achieves one of the $2k$ largest magnitudes among the entries of $\mathbf{x}$.

Proof. If $2k \geq N$, then this is trivially true. Therefore, we assume that $2k+1 \leq N$. Consider reordering the entries of $\mathbf{x}$ such that $|x_{j_1}| \geq \cdots \geq |x_{j_N}|$. Then we have

$$|x_n| > \frac{1}{k}\,\sigma_k(\mathbf{x})_1 = \frac{1}{k}\sum_{\ell=k+1}^{N} |x_{j_\ell}| \geq \frac{1}{k}\sum_{\ell=k+1}^{2k} |x_{j_\ell}| \geq |x_{j_{2k+1}}|\,.$$

It follows that the entry x_n is one of the $2k$ largest magnitudes among the entries of $\mathbf{x}$. □

Remark 3.2.2. Lemma 3.2.1 shows that for each vector $\mathbf{x} \in \mathbb{R}^N$, we have $O(k)$ entries of $\mathbf{x}$ that satisfy (3.1). As for the lower bound, we note that if $\mathbf{x}$ has exactly k nonzero entries, then these k nonzero entries satisfies (3.1). Therefore, the bound is tight in general.

Theorem 3.2.3. Let $\mathbf{x}$ and $\mathbf{z} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let $S \subseteq [N]$ be an index set containing a subset S_{2k} of size $\min(2k, |S|)$ such that the entries of $\mathbf{x}$ corresponding to S_{2k} are of largest magnitude. Suppose that

$$\mathbf{z}_S = \mathbf{z} \quad (\text{so that } z_n = 0 \text{ for each } n \notin S), \tag{3.2}$$

and that

$$|x_n - z_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1 \quad \text{for each } n \in S. \tag{3.3}$$

Then applying Algorithm 5 with inputs $\mathbf{z}$, k and S produces an output $\widehat{\mathbf{x}}$ satisfying

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \frac{1 + 4\sqrt{2}}{\sqrt{k}}\,\sigma_k(\mathbf{x})_1,$$

and the runtime of Algorithm 5 is $O\left(|S| \log |S|\right)$.

Proof. We first establish the runtime. Line 2 of Algorithm 5 can be computed in $O(|S| \log |S|)$ time, and Line 3–5 in $O(|S|)$ time.

Now let $\delta = \frac{1}{k}\sigma_k(\mathbf{x})_1$. We shall show that applying Algorithm 5 with inputs $\mathbf{z}$ and S produces an output $\widehat{\mathbf{x}}$ satisfying

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \sigma_{2k}(\mathbf{x})_2 + 4\sqrt{2k}\delta, \tag{3.4}$$

from which the desired error guarantee follows using the known inequality [FR13, Proposition 2.3]

$$\sigma_{2k}(\mathbf{x})_2 = \sigma_k(\mathbf{x} - \mathbf{x}_k)_2 \leq \frac{1}{\sqrt{k}}\,\|\mathbf{x} - \mathbf{x}_k\|_1 = \frac{1}{\sqrt{k}}\,\sigma_k(\mathbf{x})_1 = \sqrt{k}\delta.$$

With reference to Algorithm 5, we have

$$\begin{aligned}
\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 &= \|\mathbf{x} - \mathbf{z}_{\widetilde{S}}\|_2 \\
&\leq \left\|\mathbf{x} - \mathbf{x}_{\widetilde{S}}\right\|_2 + \left\|\mathbf{x}_{\widetilde{S}} - \mathbf{z}_{\widetilde{S}}\right\|_2 \\
&= \left\|\mathbf{x} - \mathbf{x}_{\widetilde{S}}\right\|_2 + \sqrt{\sum_{n \in \widetilde{S}} |x_n - z_n|^2} \\
&\leq \left\|\mathbf{x} - \mathbf{x}_{\widetilde{S}}\right\|_2 + \sqrt{2k}\delta,
\end{aligned}$$

where the last inequality follows from $\left|\widetilde{S}\right| \leq 2k$ and (3.3) for $n \in \widetilde{S} \subseteq S$.

To show (3.4), it is therefore sufficient to demonstrate that

$$\left\|\mathbf{x} - \mathbf{x}_{\widetilde{S}}\right\|_2 \leq \sigma_{2k}(\mathbf{x})_2 + 3\sqrt{2k}\delta.$$

Since

$$\begin{aligned}\left\|\mathbf{x} - \mathbf{x}_{\widetilde{S}}\right\|_2 &= \sqrt{\sum_{n \notin \widetilde{S}} |x_n|^2} \\ &\leq \sqrt{\sum_{n \notin S_{2k}} |x_n|^2 + \sum_{n \in S_{2k} \setminus \widetilde{S}} |x_n|^2} \\ &\leq \|\mathbf{x} - \mathbf{x}_{S_{2k}}\|_2 + \sqrt{\sum_{n \in S_{2k} \setminus \widetilde{S}} |x_n|^2} \\ &\leq \sigma_{2k}(\mathbf{x})_2 + \sqrt{\sum_{n \in S_{2k} \setminus \widetilde{S}} |x_n|^2},\end{aligned}$$

it is sufficient to show that

$$\sqrt{\sum_{n \in S_{2k} \setminus \widetilde{S}} |x_n|^2} \leq 3\sqrt{2k}\delta. \tag{3.5}$$

To show (3.5), we may assume that $S_{2k} \setminus \widetilde{S}$ is nonempty as otherwise the left-hand side is 0 and (3.5) holds trivially. Then $\widetilde{S} \setminus S_{2k}$ is nonempty because $|\widetilde{S}| = |S_{2k}| = \min(2k, |S|)$. We shall show that for each $n \in S_{2k} \setminus \widetilde{S}$, and for each $j \in \widetilde{S} \setminus S_{2k}$,

$$|x_n| \leq |z_n| + \delta \leq |z_j| + \delta \leq |x_j| + 2\delta \leq 3\delta. \tag{3.6}$$

It follows that

$$\sqrt{\sum_{n \in S_{2k} \setminus \widetilde{S}} |x_n|^2} \leq 3\delta\sqrt{|S_{2k}|} \leq 3\sqrt{2k}\delta,$$

which establishes (3.5). The first and third inequalities of (3.6) follows from (3.3) for $n \in S_{2k} \subseteq S$ and for $j \in \widetilde{S} \subseteq S$. The last inequality of (3.6) follows from $j \notin S_{2k}$ and the contrapositive of Lemma 3.2.1 which imply $|x_j| \leq \delta$. It remains to show

$$|z_n| \leq |z_j|,$$

which gives the second inequality of (3.6). Note that since $S_{2k} \setminus \widetilde{S}$ is nonempty and $S_{2k} \subseteq S$, we have $\widetilde{S} \neq S$. It follows that in Algorithm 5, Line 5 is not executed, therefore the condition in Line 1 is satisfied, and so Line 2 is executed. Since $j \in \widetilde{S}$ and $n \in \left(S_{2k} \setminus \widetilde{S}\right) \subseteq \left(S \setminus \widetilde{S}\right)$, we must have $|z_j| \geq |z_n|$. □

In Theorem 3.2.3, S is required to contain S_{2k}, a set of $\min(2k, |S|)$ indices for which the corresponding entries of $\mathbf{x}$ are of largest magnitude such that $\mathbf{z}$ satisfies (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. We now allow S to omit some of the indices of S_{2k}, provided that the magnitude of all corresponding entries of $\mathbf{x}$ that are retained is large compared with the magnitude of those that are omitted.

Corollary 3.2.4. Let $\mathbf{x}, \mathbf{z} \in \mathbb{R}^N$, let $k \in [N]$ be nonzero, and let $S \subseteq [N]$ contain the index set

$$T := \left\{ n \in [N] : |x_n| > \frac{1}{k}\,\sigma_k(\mathbf{x})_1 \right\}. \tag{3.7}$$

Suppose that $\mathbf{z}$ satisfies (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. Then applying Algorithm 5 to inputs $\mathbf{z}$, k, and S produces an output $\widehat{\mathbf{x}}$ satisfying

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \frac{1 + 4\sqrt{2}}{\sqrt{k}}\,\sigma_k(\mathbf{x})_1,$$

and the runtime of Algorithm 5 is $O\left(|S| \log |S|\right)$.

Proof. The runtime bound is established as in the proof of Theorem 3.2.3. It remains to show the error guarantee. Let S_{2k} be a set of $\min(2k, N)$ indices for which the corresponding entries of $\mathbf{x}$ are of largest magnitude. If $S_{2k} \subseteq S$, then we are done by Theorem 3.2.3. Otherwise, S omits some of the indices in S_{2k}, so that $S_{2k} \setminus S$ is nonempty.

Consider the set $S' = S \cup S_{2k} = S \cup (S_{2k} \setminus S)$. We first observe that applying Theorem 3.2.3 gives the same output $\widehat{\mathbf{x}}$ for inputs $\mathbf{z}$ and S as it does for inputs $\mathbf{z}$ and S'. This follows from $z_n = 0$ for $n \in S_{2k} \setminus S$. We shall show that $\mathbf{z}$ satisfies

$$z_n = 0 \quad \text{for each } n \notin S' \tag{3.8}$$

and

$$|x_n - z_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1 \quad \text{for each } n \in S', \tag{3.9}$$

so that we can apply Theorem 3.2.3 with inputs $\mathbf{z}$ and S' to obtain the desired error guarantee for $\widehat{\mathbf{x}}$. To show (3.8), use $S' \supseteq S$ and the assumption that $\mathbf{z}$ satisfies (3.2) with respect to S. To show (3.9), by the assumption that $\mathbf{z}$ satisfies (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S, it is sufficient to show $|x_n - z_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1$ for each $n \in S' \setminus S$. Let $n \notin S$. Then $n \notin T$ because S contains T, so $|x_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1$ by (3.7). Also $z_n = 0$ by (3.2). Therefore $|x_n - z_n| = |x_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1$, as required. □

3.3 Estimation Scheme

In this section, we describe the estimation phase depicted in Figure 3.1. This provides an estimation scheme $(\mathcal{M}_{\text{est}}, \Delta_{\text{est}})$ for estimating entries of a vector $\mathbf{x} \in \mathbb{R}^N$ using non-adaptive linear measurements $\mathcal{M}_{\text{est}}\, \mathbf{x}$ and estimation algorithm Δ_{est}. This scheme is of interest in its own right.

We shall show that the combinatorial structure of (K, α)-coherent matrices can be used to estimate the entries of $\mathbf{x}$ at an arbitrary index set $S \subseteq [N]$ with high accuracy as in (3.3), namely to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$. In particular, by taking S to be an index set containing the index set T of (3.7), we obtain a vector $\mathbf{z}$ satisfying (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. By Corollary 3.2.4, $\mathbf{z}$ can be used as an input to Algorithm 5 to produce an approximation $\widehat{\mathbf{x}}$ to $\mathbf{x}$ with an error guarantee of

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \frac{1 + 4\sqrt{2}}{\sqrt{k}}\, \sigma_k(\mathbf{x})_1.$$

Our main results are the estimation schemes in Corollaries 3.3.8 and 3.3.10, whose number of measurements are $\Theta(k^2 \log N)$ and $\Theta(k^2 \log_k^2 N)$ respectively. These schemes use Algorithm 6 as the recovery algorithm. We begin with two lemmas that are introduced solely to prove Theorem 3.3.4, from which Corollary 3.3.5 is derived.

Definition 3.3.1. Let $A \in \{0,1\}^{m \times N}$ be a binary matrix. For each $n \in [N]$, write $A(n)$ for the submatrix of A comprising the rows of A whose entry in column n is 1, and write $A'(n)$ for the submatrix of $A(n)$ obtained by deleting column n.

Lemma 3.3.2. Let $\mathcal{M}_{\text{C}}$ be a (K, α)-coherent matrix with N columns. Let $n \in [N]$, let $k \in [1, K/\alpha]$ be an integer, and let $\mathbf{x} \in \mathbb{R}^{N-1}$ be nonzero. Then at most $k\alpha$ of the K entries of $\mathcal{M}'_{\text{C}}(n)\, \mathbf{x}$ have magnitude greater than or equal to $\frac{1}{k}\|\mathbf{x}\|_1$.

Proof. Since $\mathbf{x}$ is nonzero, we may write $A_j = \dfrac{k}{\|\mathbf{x}\|_1}\left|\left(\mathcal{M}'_{\text{C}}(n)\, \mathbf{x}\right)_j\right|$. We then have that

$$\begin{aligned}
&\#j : \ \left|\left(\mathcal{M}'_{\text{C}}(n)\, \mathbf{x}\right)_j\right| \geq \frac{1}{k}\|\mathbf{x}\|_1 \\
=\ &\#j : \ A_j \geq 1 \\
\leq\ &\sum_j |A_j| \\
=\ &\frac{k}{\|\mathbf{x}\|_1} \sum_j \left|\left(\mathcal{M}'_{\text{C}}(n)\, \mathbf{x}\right)_j\right| \\
=\ &k \frac{\|\mathcal{M}'_{\text{C}}(n)\, \mathbf{x}\|_1}{\|\mathbf{x}\|_1}.
\end{aligned}$$

It remains to show that $\frac{\left\|\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{x}\right\|_1}{\|\mathbf{x}\|_1} \le \alpha$. This follows from

$$\begin{aligned}
\sup\left\{\frac{\|\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{x}\|_1}{\|\mathbf{x}\|_1} : \mathbf{x} \in \mathbb{R}^{N-1}, \mathbf{x} \ne \mathbf{0}\right\} =& \|\mathcal{M}'_{\mathrm{C}}(n)\|_1 \\
=& \max_j \left\|\text{column } j \text{ of } \mathcal{M}'_{\mathrm{C}}(n)\right\|_1 \\
=& \max_j \left(\#\ 1\text{s in column } j \text{ of } \mathcal{M}'_{\mathrm{C}}(n)\right) \\
=& \max_{j \ne n} \left(\text{dot product of columns } j \text{ and } n \text{ of } \mathcal{M}_{\mathrm{C}}\right) \\
\le& \alpha
\end{aligned}$$

because $\mathcal{M}_{\mathrm{C}}$ is a (K, α)-coherent matrix. □

We now prove our second lemma. Recall the following notation from Section 3.2: given $\mathbf{x} = (x_i) \in \mathbb{R}^N$ and a subset $S \subseteq [N]$, the vector $\mathbf{x}_S \in \mathbb{R}^N$ is given by

$$(x_S)_i = \begin{cases} x_i & \text{if } i \in S, \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 3.3.3. Let $\mathcal{M}_{\mathrm{C}}$ be a (K, α)-coherent matrix with N columns. Let $n \in [N]$, let $k \in [1, K/\alpha]$ be an integer, $S \subseteq [N]$ with $|S| = k$, and $\mathbf{x} \in \mathbb{R}^{N-1}$. Then $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{x}$ and $\mathcal{M}'_{\mathrm{C}}(n)\ (\mathbf{x} - \mathbf{x}_S)$ differ in at most $k\alpha$ of their K entries.

Proof. If $\mathbf{x} = \mathbf{0}$, then we are done. We may therefore assume that $\mathbf{x} \ne \mathbf{0}$. Let $\mathbb{1} \in \mathbb{R}^{N-1}$ be the vector of all ones. We have that

$$\begin{aligned}
& \#j : \ \left(\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{x}\right)_j \ne \left(\mathcal{M}'_{\mathrm{C}}(n)\,(\mathbf{x} - \mathbf{x}_S)\right)_j \\
=\ & \#j : \ \left(\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{x}_S\right)_j \ne 0 \\
\le\ & \#j : \ \left(\mathcal{M}'_{\mathrm{C}}(n)\,\mathbb{1}_S\right)_j \ge 1 \\
=\ & \#j : \ \left(\mathcal{M}'_{\mathrm{C}}(n)\,\mathbb{1}_S\right)_j \ge \frac{1}{k}\|\mathbb{1}_S\|_1 \\
\le\ & k\alpha,
\end{aligned}$$

where the first inequality holds because all nonzero entries of $\mathcal{M}'_{\mathrm{C}}(n)$ are 1, and the last inequality follows from applying Lemma 3.3.2 with $\mathbf{x} = \mathbb{1}_S$. □

By combining the two lemmas above, we are able to bound the accuracy with which we can approximate each entry of a vector $\mathbf{x} \in \mathbb{R}^N$ using only non-adaptive linear measurements from a (K, α)-coherent matrix.

Theorem 3.3.4. Let $\mathcal{M}_{\mathrm{C}}$ be a (K,α)-coherent matrix with N columns. Let $n \in [N]$, $\mathbf{x} \in \mathbb{R}^N$, and let $c \geq 2$ and $k \in \left[1, \frac{K}{c\alpha}\right)$ be integers. Then, for more than $\left(1 - \frac{2}{c}\right) K$ values of $j \in [K]$,

$$\left|\left(\mathcal{M}_{\mathrm{C}}(n)\,\mathbf{x}\right)_j - x_n\right| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1.$$

That is, x_n is estimated to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$ by a proportion larger than $1 - \frac{2}{c}$ of the K entries of $\mathcal{M}_{\mathrm{C}}(n)\,\mathbf{x}$.

Proof. If $\mathbf{x} = \mathbf{0}$, then we are done. We may therefore assume that $\mathbf{x} \neq \mathbf{0}$. Let $\mathbf{w} = (x_0, x_1, \ldots, x_{n-1}, x_{n+1}, \cdots, x_{N-1}) \in \mathbb{R}^{N-1}$. For each $j \in [K]$, we have

$$\begin{aligned}
\left(\mathcal{M}_{\mathrm{C}}(n)\,\mathbf{x}\right)_j - x_n &= \sum_{i=1}^{N} \left(\mathcal{M}_{\mathrm{C}}\,(n)\right)_{j,i} x_i - x_n \\
&= \sum_{i \neq n} \left(\mathcal{M}_{\mathrm{C}}\,(n)\right)_{j,i} x_i + \left(\mathcal{M}_{\mathrm{C}}\,(n)\right)_{j,n} x_n - x_n \\
&= \left(\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}\right)_j
\end{aligned}$$

because $\left(\mathcal{M}_{\mathrm{C}}\,(n)\right)_{j,n} = 1$. It is therefore sufficient to show that

$$\left|\left(\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}\right)_j\right| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1 \quad \text{for more than } \left(1 - \tfrac{2}{c}\right) K \text{ values of } j \in [K]. \tag{3.10}$$

We claim firstly that at most $k\alpha$ entries of $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ differ from their corresponding entry in $\mathcal{M}'_{\mathrm{C}}(n)\,(\mathbf{w} - \mathbf{w}_k)$. In the case that $\mathbf{w} - \mathbf{w}_k = \mathbf{0}$, this gives the required result because this claim shows that at most $k\alpha < \frac{K}{c}$ entries of $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ are nonzero and so more than $(1 - \frac{1}{c})K$ entries of $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ are zero (and therefore (3.10) holds trivially). Otherwise, in the case that $\mathbf{w} - \mathbf{w}_k \neq \mathbf{0}$, we claim secondly that at most $k\alpha$ of the remaining (at least) $K - k\alpha$ entries have magnitude greater than or equal to $\frac{1}{k}\,\sigma_k(\mathbf{w})_1$. It then follows that at most $2k\alpha < \frac{2K}{c}$ entries of $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ have magnitude greater than or equal to $\frac{1}{k}\,\sigma_k(\mathbf{w})_1$, so that more than $\left(1 - \frac{2}{c}\right) K$ entries of $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ have magnitude bounded by

$$\frac{1}{k}\,\sigma_k(\mathbf{w})_1 \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1,$$

giving (3.10).

We now prove the first claim. Since $k < \frac{K}{c\alpha} < \frac{K}{\alpha}$, we may apply Lemma 3.3.3 where S is an index set for k of the largest entries of $\mathbf{w}$ to obtain that $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ and $\mathcal{M}'_{\mathrm{C}}(n)\,(\mathbf{w} - \mathbf{w}_k)$ differ in at most $k\alpha$ of their K entries, as required.

We now prove the second claim. All remaining entries of $\mathcal{M}'_{\mathrm{C}}(n)\,\mathbf{w}$ are equal to their corresponding entry in $\mathcal{M}'_{\mathrm{C}}(n)\,(\mathbf{w} - \mathbf{w}_k)$, so it is sufficient to prove that at most $k\alpha$ entries of

$\mathcal{M}'_{\mathrm{C}}(n)\,(\mathbf{w}-\mathbf{w}_k)$ have magnitude greater than or equal to $\frac{1}{k}\,\|\mathbf{w}-\mathbf{w}_k\|_1$. This is given by replacing $\mathbf{x}$ in Lemma 3.3.2 by $\mathbf{w}-\mathbf{w}_k$, which by assumption is nonzero. □

We now apply Theorem 3.3.4 in the special case $c = 4$, and use $\mathcal{M}_{\mathrm{C}}$ as the estimation matrix $\mathcal{M}_{\mathrm{est}}$. This will allow us to estimate x_n accurately for each n in an arbitrary index set $S \subseteq [N]$, using the median of the entries of the vector $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$.

Corollary 3.3.5. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let $\mathcal{M}_{\mathrm{est}} = \mathcal{M}_{\mathrm{C}} \in \{0,1\}^{t\times N}$ be a (K,α)-coherent matrix where $K > 4k\alpha$. Let $S \subseteq [N]$. Then for each index $n \in S$, the median of the entries of the vector $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$ estimates x_n to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$.

Proof. Fix an index $n \in S$ and take $c = 4$ in Theorem 3.3.4. This shows that more than half of the K entries of $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$ estimate x_n to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$, and so the median of the entries of $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$ estimates x_n to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$. □

Remark 3.3.6. Note that the vector $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$ can be computed from $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ by selecting entries according to the locations of the 1 entries in column n of $\mathcal{M}_{\mathrm{est}}$. This is used in Line 3 of Algorithm 6 below. Consequently, we must either store the locations of the 1 entries of each column of $\mathcal{M}_{\mathrm{est}}$, or else find an efficient way to determine the locations of these 1 entries: see Corollary 3.3.10.

We combine the ideas in this section to give an estimation scheme $(\mathcal{M}_{\mathrm{est}}, \Delta_{\mathrm{est}})$, where $\mathcal{M}_{\mathrm{est}} = \mathcal{M}_{\mathrm{C}}$ is a (K,α)-coherent matrix and Δ_{est} is Algorithm 6. The performance of this scheme is given in Theorem 3.3.7.

Algorithm 6 Median-based Estimation [BIS12, Algorithm 1]

```
   Input: $S \subseteq [N]$ and $\mathcal{M}_{\mathrm{est}} \in \{0,1\}^{t\times N}$ and $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$
   Output: $\mathbf{z} = (z_n) \in \mathbb{R}^N$
1: Initialize $\mathbf{z} \leftarrow \mathbf{0} \in \mathbb{R}^N$
2: for each $n$ in $S$ do
3:     $z_n \leftarrow$ median of the entries of $\mathbf{y}_{\mathrm{est}}$ corresponding to 1 entries in column $n$ of $\mathcal{M}_{\mathrm{est}}$
4:                      ▷ $z_n$ is the median of the entries of $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$, see Remark 3.3.6
5: end for
6: Output: $\mathbf{z}$
```

Theorem 3.3.7. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let the estimation matrix $\mathcal{M}_{\mathrm{est}} = \mathcal{M}_{\mathrm{C}} \in \{0,1\}^{t\times N}$ be a (K,α)-coherent matrix, where $K > 4k\alpha$. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ be the estimation measurement, and let $S \subseteq [N]$. Then applying Algorithm 6 to inputs S, $\mathcal{M}_{\mathrm{est}}$, $\mathbf{y}_{\mathrm{est}}$ produces an output $\mathbf{z}$ satisfying (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S.

Assuming that the locations of the 1 entries in column n of $\mathcal{M}_{\mathrm{C}}$ are stored for each n, the runtime of Algorithm 6 is $\Theta\left(|S|K\right)$.

Proof. It follows from Line 1 and Lines 2–5 of Algorithm 6 that $z_n = 0$ for each $n \notin S$, hence (3.2) holds. By Line 4 of Algorithm 6 and Corollary 3.3.5, we have $|x_n - z_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1$ for each $n \in S$ and so (3.3) holds.

We now bound the runtime. For each $n \in S$, the locations of the K '1' entries in column n of $\mathcal{M}_{\mathrm{est}}$ are stored, and so the median in Line 3 can be computed in $\Theta(K)$ time using the median-of-medians algorithm [BFP+73]. Therefore, the runtime of Lines 2–5 is $\Theta(|S|K)$. □

To derive a good estimation scheme from Theorem 3.3.7, it remains to identify a (K,α)-coherent matrix $\mathcal{M}_{\mathrm{C}} \in \{0,1\}^{t\times N}$ with $K > 4k\alpha$ and a small row count t. This is provided by taking a Porat-Rothschild (K,α)-coherent matrix from Theorem 2.4.7 where $\alpha = \Theta\left(\log N\right)$ and $K = 4k\alpha + 1 = \Theta(k \log N)$, whose row count is $t = \Theta(K^2/\alpha) = \Theta(k^2 \log N)$. This gives us an estimation scheme with the following performance.

Corollary 3.3.8. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let the estimation matrix $\mathcal{M}_{\mathrm{est}} = \mathcal{M}_{\mathrm{PR}} \in \{0,1\}^{t\times N}$ be a Porat-Rothschild (K,α)-coherent matrix, where $\alpha = \Theta\left(\log N\right)$ and $K = 4k\alpha + 1$. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ be the estimation measurement, and let $S \subseteq [N]$. Then applying Algorithm 6 to inputs S, $\mathcal{M}_{\mathrm{est}}$, and $\mathbf{y}_{\mathrm{est}}$ produces an output $\mathbf{z}$ satisfying (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. Assuming that the locations of the 1 entries of column n of $\mathcal{M}_{\mathrm{est}}$ are stored for each n, the runtime of Algorithm 6 is $\Theta\left(|S|\,k \log N\right)$. Furthermore, $t = \Theta(k^2 \log N)$.

Remark 3.3.9. A weakness of the estimation scheme in Corollary 3.3.8 is that the memory required to store the locations of the 1 entries of the columns of $\mathcal{M}_{\mathrm{est}}$ is $\Omega(KN) = \Omega(Nk\log N)$. We now provide a low-memory variant of Corollary 3.3.8 which allows us to determine the locations of the 1 entries in each column of $\mathcal{M}_{\mathrm{est}}$ "on the fly" instead of storing them, by taking advantage of the strongly explicit construction of a Kautz-Singleton (K,α)-coherent matrix $\mathcal{M}_{\mathrm{KS}} \in \{0,1\}^{t\times N}$ from Theorem 2.4.6 where $\alpha = \log_k N$ and $K = 4k\alpha + 1 = \Theta(k \log_k N)$. Note that since $K > k$, we have $\alpha = \log_k N > \log_K N$ and so the condition on α in the statement of Theorem 2.4.6 holds. Furthermore, the row count of $\mathcal{M}_{\mathrm{KS}}$ is $t = \Theta(K^2) = \Theta(k^2 \log_k^2 N)$.

Corollary 3.3.10. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let the estimation matrix $\mathcal{M}_{\mathrm{est}} = \mathcal{M}_{\mathrm{KS}} \in \{0,1\}^{t\times N}$ be a Kautz-Singleton (K,α)-coherent matrix, where $\alpha = \log_k N$ and $K = 4k\alpha + 1$. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ be the estimation measurement, and let $S \subseteq [N]$. Then applying Algorithm 6 to inputs S, $\mathcal{M}_{\mathrm{est}}$, and $\mathbf{y}_{\mathrm{est}}$ produces an output $\mathbf{z}$ satisfying (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. The runtime of Algorithm 6 is $O\left(|S|\,k \log_k^2 N\right)$. Furthermore, $t = \Theta(k^2 \log_k^2 N)$.

Proof. The only modification to the scheme described in Corollary 3.3.8 is that Line 3 of Algorithm 6 now determines the locations of the 1 entries of column n of $\mathcal{M}_{\text{est}}$. Each of the $K = \Theta(k \log_k N)$ row blocks contains exactly one 1 entry in column n of $\mathcal{M}_{\text{est}}$, and locating each of them takes $O(\log_K N)$ time by Theorem 2.4.6. Therefore, each instance of Line 3 takes $O(K \log_K N)$ time and so the runtime is $O(|S| K \log_K N) = O\left(|S|\, k \log_k^2 N\right)$, using $K = \Omega(k)$. The row count t of $\mathcal{M}_{\text{KS}}$ follows immediately from Theorem 2.4.6. □

The estimation schemes of Corollaries 3.3.8 and 3.3.10 allow us to estimate the entries of $\mathbf{x}$ at an arbitrary index set $S \subseteq [N]$. In particular, when the index set S contains the set T defined in (3.7), by Corollary 3.2.4 we obtain an approximation $\widehat{\mathbf{x}}$ satisfying an ℓ_2/ℓ_1 error guarantee. The condition that S contains T can be trivially satisfied by taking $S = [N]$. This gives the following Corollaries 3.3.11 and 3.3.12 (variants of Corollaries 3.3.8 and 3.3.10, respectively), each of which describes an identification-free scheme (see Algorithm 4 and the discussion preceding it) consisting of only an estimation matrix, estimation algorithm, and pruning algorithm (Algorithm 5).

Corollary 3.3.11. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let the estimation matrix $\mathcal{M}_{\text{est}} = \mathcal{M}_{\text{PR}} \in \{0,1\}^{t \times N}$ be a Porat-Rothschild (K, α)-coherent matrix, where $\alpha = \Theta(\log N)$ and $K = 4k\alpha + 1$. Let $\mathbf{y}_{\text{est}} = \mathcal{M}_{\text{est}}\, \mathbf{x}$ be the estimation measurement. Let $\mathbf{z}$ be the output of Algorithm 6 with inputs $S = [N]$, $\mathcal{M}_{\text{est}}$, and $\mathbf{y}_{\text{est}}$. Then applying Algorithm 5 to inputs $\mathbf{z}$, k and $S = [N]$ produces an output $\widehat{\mathbf{x}}$ satisfying

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \le \frac{1 + 4\sqrt{2}}{\sqrt{k}}\, \sigma_k(\mathbf{x})_1.$$

Assuming that the locations of the 1 entries of column n of $\mathcal{M}_{\text{est}}$ are stored for each n, the total runtime of Algorithms 6 and 5 is $O(kN \log N)$. Furthermore, $t = O(k^2 \log N)$.

Corollary 3.3.12. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let the estimation matrix $\mathcal{M}_{\text{est}} = \mathcal{M}_{\text{KS}} \in \{0,1\}^{t \times N}$ be a Kautz-Singleton (K, α)-coherent matrix, where $\alpha = \log_k N$ and $K = 4k\alpha + 1$. Let $\mathbf{y}_{\text{est}} = \mathcal{M}_{\text{est}}\, \mathbf{x}$ be the estimation measurement. Let $\mathbf{z}$ be the output of Algorithm 6 with inputs $S = [N]$, $\mathcal{M}_{\text{est}}$, and $\mathbf{y}_{\text{est}}$. Then applying Algorithm 5 to inputs $\mathbf{z}$ and $S = [N]$ produces an output $\widehat{\mathbf{x}}$ satisfying

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \le \frac{1 + 4\sqrt{2}}{\sqrt{k}}\, \sigma_k(\mathbf{x})_1.$$

The total runtime of Algorithms 6 and 5 is $O\left(kN \log_k^2 N\right)$. Furthermore, $t = O(k^2 \log_k^2 N)$.

The compressed sensing schemes of Corollaries 3.3.11 and 3.3.12 satisfy the condition $S \supseteq T$ by taking $S = [N]$, but at the cost of an $\Omega(N)$ runtime (see Line 2 of Algorithm 6). In the next section, we show instead how to efficiently identify a relatively small set S containing T, from which Algorithm 6 then gives a good approximation $\widehat{\mathbf{x}}$ quickly.

3.4 Identification Scheme

In this section, we describe the identification phase depicted in Figure 3.1. This provides an identification scheme $(\mathcal{M}_{\text{id}}, \Delta_{\text{id}})$ for identifying the largest entries of a vector $\mathbf{x} \in \mathbb{R}^N$, where $\mathcal{M}_{\text{id}}$ is an identification matrix used to produce non-adaptive linear measurements $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\, \mathbf{x}$, and Δ_{id} is an identification algorithm used to recover from $\mathcal{M}_{\text{id}}$ and $\mathbf{y}_{\text{id}}$ an index set $S \subseteq N$ containing the indices of the largest entries of $\mathbf{x}$. This identification scheme is also of independent interest.

We call each index $n \in [N]$ of $\mathbf{x} = (x_j)$ satisfying

$$|x_n| > \frac{1}{k}\, \sigma_k(\mathbf{x})_1 \tag{3.11}$$

a heavy hitter (of $\mathbf{x}$ with respect to k). The set of heavy hitters is the set T specified in (3.7). As explained at the end of Section 3.3, we wish to quickly identify a relatively small set $S \subseteq [N]$ containing every heavy hitters. We shall show that this identification can be carried out efficiently using the columnwise Kronecker product (see Definition 3.4.1) of a (K, α)-coherent matrix and a bit-test matrix (see Definition 3.4.2).

Our main result is Corollary 3.4.16, which provides an identification scheme whose row count and runtime are both $O(k^2 \log^2 N)$. Corollary 3.4.16 is a consequence of Theorem 3.4.15, in which we apply the key technique of the "bit-testing" to identify all heavy hitters. We demonstrate the use of this technique on a *single* fixed heavy hitter in Corollary 3.4.6. A heavy hitter n that has a "good" row vector (one satisfying (3.15)) can be identified according to Theorem 3.4.8, and Remark 3.4.9 describes how to extend this procedure to identify *all* the heavy hitters using a single (K, α)-coherent matrix. In particular, Corollary 3.4.11 shows that a suitable (K, α)-coherent matrix contains more than $K/3$ good rows with respect to each heavy hitter. This leads to Theorem 3.4.12, which identifies a multiset U of indices guaranteed to contain every heavy hitter. Remark 3.4.14 then shows how to remove *some* of the indices in U which are certainly not heavy hitters. We then present a general identification scheme in Theorem 3.4.15, whose performance depends on the (K, α)-coherent matrix used. By applying this to a Porat-Rothschild (K, α)-coherent matrix from Theorem 2.4.7, we obtain Corollary 3.4.16.

We first introduce some notation and definitions. For a matrix A whose rows are $\mathbf{a}_1, \ldots, \mathbf{a}_m$, we write $A = (\mathbf{a}_i)$.

Definition 3.4.1. Let $\mathcal{R} = (\mathbf{r}_\ell)$ be a $t \times N$ matrix and let $\mathcal{B} = (\mathbf{b}_i)$ be an $b \times N$ matrix. The columnwise Kronecker product of $\mathcal{R}$ and $\mathcal{B}$, denoted $\mathcal{R} \circledast \mathcal{B}$, is the $tb \times N$ matrix comprising t blocks of b rows each, where the i^{th} row of the ℓ^{th} block is the entrywise product of $\mathbf{r}_\ell$ and $\mathbf{b}_i$.

Definition 3.4.2. The N^{th} bit-test matrix $\mathcal{B}_N \in \{0,1\}^{(1+\lceil \log_2 N \rceil)\times N}$ is the binary matrix whose column j (read from top to bottom) equals 1 followed by the binary representation of j.

Example 3.4.3. Let

$$\mathcal{R} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}.$$

Then $\mathcal{B}$ is the 4^{th} bit-test matrix $\mathcal{B}_4$, and

$$\mathcal{R} \circledast \mathcal{B} = \left[\begin{array}{cccc} 1 & 2 & 3 & 4 \\ 0 & 0 & 3 & 4 \\ 0 & 2 & 0 & 4 \\ \hdashline 5 & 6 & 7 & 8 \\ 0 & 0 & 7 & 8 \\ 0 & 6 & 0 & 8 \end{array}\right].$$

Remark 3.4.4. Let $x \in \mathbb{R}^N$. Let $\mathcal{R} = (\mathbf{r}_\ell)$ be a $t \times N$ matrix, and let $\mathcal{B}_N = (\mathbf{b}_i)$ be the N^{th} bit-test matrix. The vector $\mathbf{y} = (y_j) = (\mathcal{R} \circledast \mathcal{B}_N)\mathbf{x} \in \mathbb{R}^{t(1+\lceil \log_2 N \rceil)}$ comprises t blocks of $1 + \lceil \log_2 N \rceil$ rows each, the ℓ^{th} block being $(\mathbf{r}_\ell \circledast \mathcal{B}_N)\mathbf{x} \in \mathbb{R}^{1+\lceil \log_2 N \rceil}$. Furthermore, for each $\ell \in [t]$ and $i \in [1 + \lceil \log_2 N \rceil]$, the i^{th} row of the ℓ^{th} block of $\mathbf{y}$ is

$$y_{\ell(1+\lceil \log_2 N \rceil)+i} = \left((\mathbf{r}_\ell \circledast \mathcal{B}_N)\, \mathbf{x} \right)_i = \langle \mathbf{r}_\ell \circledast \mathbf{b}_i, \mathbf{x} \rangle. \tag{3.12}$$

Since $\mathbf{b}_0$ is a row vector of all 1s, $\mathbf{r}_\ell \circledast \mathbf{b}_0 = \mathbf{r}_\ell$ for each $\ell \in [t]$. Applying (3.12) with $i = 0$ shows that the ℓ^{th} entry of $\mathcal{R}\mathbf{x} \in \mathbb{R}^t$ is stored in the 0^{th} row of the ℓ^{th} block of $\mathbf{y}$:

$$y_{\ell(1+\lceil \log_2 N \rceil)} = \langle \mathbf{r}_\ell \circledast \mathbf{b}_0, \mathbf{x} \rangle = \langle \mathbf{r}_\ell, \mathbf{x} \rangle = (\mathcal{R}\mathbf{x})_\ell\,. \tag{3.13}$$

We will use (3.12) and (3.13) in Line 4 of Algorithm 7, as well as in the proofs of Theorem 3.4.8 and 3.4.12.

Lemma 3.4.5. Let $\mathbf{x} \in \mathbb{R}^N$ and $n \in [N]$, and let $\mathbf{b}, \mathbf{r} \in \{0,1\}^N$ be row vectors. Suppose that

$$r_n = 1 \quad \text{and} \quad |x_n| > \sum_{j \neq n} r_j |x_j|. \tag{3.14}$$

Then

$$|\langle \mathbf{r} \circledast \mathbf{b}, \mathbf{x}\rangle| > |\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}, \mathbf{x}\rangle| \quad \Longleftrightarrow \quad b_n = 1.$$

Proof. We will first show the case when $b_n = 1$. We have

$$\begin{aligned}
|\langle \mathbf{r} \circledast \mathbf{b}, \mathbf{x}\rangle| &= \left|\sum_j r_j b_j x_j\right| \\
&= \left|x_n + \sum_{j\neq n} r_j b_j x_j\right| \\
&\geq |x_n| - \sum_{j\neq n} r_j b_j |x_j| \\
&> \sum_{j\neq n} r_j |x_j| - \sum_{j\neq n} r_j b_j |x_j| \\
&= \sum_{j\neq n} r_j (1 - b_j)\, |x_j| \\
&= \sum_j |r_j\, (1 - b_j)\, x_j| \\
&\geq \left|\sum_j (r_j - r_j b_j) x_j\right| \\
&= |\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}, \mathbf{x}\rangle|.
\end{aligned}$$

The case when $b_n = 0$ can be obtained by applying the inequalities above to the binary vector $\mathbf{b}'$ with entries $b'_j = 1 - b_j$ such that $b'_n = 1$, giving

$$|\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}, \mathbf{x}\rangle| = |\langle \mathbf{r} \circledast \mathbf{b}', \mathbf{x}\rangle| > |\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}', \mathbf{x}\rangle| = |\langle \mathbf{r} \circledast \mathbf{b}, \mathbf{x}\rangle|\,. \qquad \square$$

We now apply Lemma 3.4.5 in the special case that

(i) $\mathbf{b}$ equals the i^{th} row $\mathbf{b}_i$ of the N^{th} bit-test matrix

(ii) n is a heavy hitter.

This will reduce the task of identifying a heavy hitter n to the task of finding a corresponding binary row vector $\mathbf{r}$ satisfying (3.15): see Remark 3.4.7.

Corollary 3.4.6. Let $\mathbf{x} \in \mathbb{R}^N$ and $k \in [N]$ be nonzero, and let $n \in [N]$ be a heavy hitter of $\mathbf{x}$ with respect to k (see (3.11)). Let $i \in \{1, 2, \ldots, \lceil \log_2 N \rceil\}$, and let $\mathbf{b}_i$ be the i^{th} row of the N^{th} bit-test matrix. Suppose that $\mathbf{r} \in \{0,1\}^N$ is a row vector satisfying

$$r_n = 1 \quad \text{and} \quad \left| \sum_j r_j |x_j| - |x_n| \right| \le \frac{1}{k}\, \sigma_k(\mathbf{x})_1. \tag{3.15}$$

Then the following bit comparison test holds:

$$|\langle \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x}\rangle| > |\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x}\rangle| \quad \iff \quad \text{the } i^{\text{th}} \text{ most significant bit of } n \text{ is } 1. \tag{3.16}$$

Proof. By (3.11) and (3.15),

$$|x_n| > \frac{1}{k}\, \sigma_k(\mathbf{x})_1 \ge \left| \sum_j r_j |x_j| - |x_n| \right| = \sum_{j \ne n} r_j |x_j|.$$

Therefore we may apply Lemma 3.4.5 with $\mathbf{b} = \mathbf{b}_i$ to deduce that

$$|\langle \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x}\rangle| > |\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x}\rangle| \quad \iff \quad \text{the } n^{\text{th}} \text{ entry of } \mathbf{b}_i \text{ is } 1.$$

By Definition 3.4.2, the n^{th} entry of $\mathbf{b}_i$ equals the i^{th} most significant bit of n. □

Remark 3.4.7. For each fixed heavy hitter n, provided we can find a good row vector $\mathbf{r} \in \{0,1\}^N$ satisfying (3.15) with respect to n and k, by Corollary 3.4.6 we can determine (the binary representation of) n from the $\lceil \log_2 N \rceil$ bit comparison tests (3.16) for $i = 1, 2, \ldots, \lceil \log_2 N \rceil$. This measurement and determination procedure is formalized below.

We write the indicator function for a condition X as

$$I[X] = \begin{cases} 1 & \text{if } X \text{ holds,} \\ 0 & \text{otherwise.} \end{cases}$$

Theorem 3.4.8. Let $\mathbf{x} \in \mathbb{R}^N$ and $k \in [N]$ be nonzero, and let $n \in [N]$ be a heavy hitter of $\mathbf{x}$ with respect to k. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix, and suppose that $\mathbf{r} \in \{0,1\}^{1 \times N}$ is a row vector satisfying (3.15) with respect to n and k. Let $\mathbf{z} = (z_i) = (\mathbf{r} \circledast \mathcal{B}_N)\, \mathbf{x} \in \mathbb{R}^{1+\lceil \log_2 N \rceil}$. Then

$$n = \sum_{i=1}^{\lceil \log_2 N \rceil} I\Big[|z_i| > |z_0 - z_i|\Big]\, 2^{\lceil \log_2 N \rceil - i}. \tag{3.17}$$

Proof. Let $\mathcal{B}_N = (\mathbf{b}_i)$. By Corollary 3.4.6, for each $i \in \{1, 2, \ldots, \lceil \log_2 N \rceil\}$ the i^{th} most significant bit of n is equal to

$$I\Big[|\langle \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x}\rangle| > |\langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x}\rangle|\Big].$$

By (3.12) with $\mathcal{R} = (\mathbf{r})$ (so $t = 1$ and $\ell = 0$) we have

$$z_i = \langle \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x} \rangle,$$

and then by (3.13) we have

$$z_0 - z_i = \langle \mathbf{r}, \mathbf{x} \rangle - \langle \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x} \rangle = \langle \mathbf{r} - \mathbf{r} \circledast \mathbf{b}_i, \mathbf{x} \rangle.$$

Therefore the i^{th} most significant bit of n is equal to $I\Big[|z_i| > |z_0 - z_i|\Big]$, so that (3.17) holds. □

Remark 3.4.9. For each heavy hitter n, it remains to find a suitable row vector $\mathbf{r}^{(n)} \in \{0,1\}^N$ satisfying (3.15) with respect to n and k. At first sight this appears to be difficult, because the condition (3.15) depends on all the entries of the unknown vector $\mathbf{x}$. However, it turns out that suitable row vectors $\mathbf{r}^{(n)}$ for every heavy hitter n are collectively contained in the rows of a single (K, α)-coherent matrix $\mathcal{M}_{\mathrm{C}}$ of size $t \times N$. We shall show in Corollary 3.4.11 that, for each heavy hitter n, more than $K/3$ of the t rows of $\mathcal{M}_{\mathrm{C}}$ satisfy (3.15). For a given heavy hitter n, we cannot determine which of the t rows satisfies (3.15), but we know there are more than $K/3$ of them. We therefore apply Theorem 3.4.8 to each of the t rows of $\mathcal{M}_{\mathrm{C}}$ in turn to produce a size t multiset U of indices n from (3.17): see Lines 2–11 of Algorithm 7. We conclude in Theorem 3.4.12 that the multiset U does not contain "false negatives": every heavy hitter is guaranteed to occur in U, and with multiplicity more than $K/3$. On the other hand, U can contain "false positives", namely indices n determined by (3.17) that are not heavy hitters. (These can arise only when $\mathbf{r}$ is a row of $\mathcal{M}_{\mathrm{C}}$ for which (3.15) fails for every heavy hitter.)

The rows of a (K, α)-coherent matrix are a good source of suitable row vectors because of the following property, which is also the key property for the median-based estimation algorithm (see Corollary 3.3.5).

Lemma 3.4.10. Let $\mathcal{M}_{\mathrm{C}} \in \{0,1\}^{t \times N}$ be a (K, α)-coherent matrix, and let $c \geq 2$ and $k \in \Big[1, \frac{K}{c\alpha}\Big)$ be integers. Let $\mathbf{x} \in \mathbb{R}^N$, and write $|\mathbf{x}| = (|x_j|)$. Then for each $n \in [N]$, there are more than $\left(1 - \frac{2}{c}\right) K$ values of $\ell \in [t]$ for which

$$(\mathcal{M}_{\mathrm{C}})_{\ell,n} = 1 \quad \text{and} \quad \Big| (\mathcal{M}_{\mathrm{C}}\, |\mathbf{x}|)_\ell - |x_n| \Big| \leq \frac{1}{k}\, \sigma_k(\mathbf{x})_1.$$

Proof. Let $n \in [N]$. By Definition 3.3.1, the values $\ell \in [t]$ satisfying the condition $(\mathcal{M}_{\mathrm{C}})_{\ell,n} = 1$ are exactly the rows of $\mathcal{M}_{\mathrm{C}}$ contained in the submatrix $\mathcal{M}_{\mathrm{C}}(n)$ of size $K \times N$. We may therefore restrict attention to $\mathcal{M}_{\mathrm{C}}(n)$, and are required to show that there are more than

$\left(1 - \frac{2}{c}\right) K$ values of $\ell \in [K]$ for which

$$\left| \left(\mathcal{M}_{\mathrm{C}}(n)\, |\mathbf{x}|\right)_\ell - |x_n| \right| \leq \frac{1}{k}\, \sigma_k(\mathbf{x})_1.$$

Apply Theorem 3.3.4 to $|\mathbf{x}|$ to show that there are more than $\left(1 - \frac{2}{c}\right) K$ values of $\ell \in [K]$ for which

$$\left| \left(\mathcal{M}_{\mathrm{C}}(n)\, |\mathbf{x}|\right)_\ell - |x_n| \right| \leq \frac{1}{k}\, \sigma_k\left(|\mathbf{x}|\right)_1 = \frac{1}{k}\, \sigma_k(\mathbf{x})_1,$$

as required. □

We now apply Lemma 3.4.10 in the special case $c = 3$. This will allow us to remove some indices falsely identified as a heavy hitter: see Remark 3.4.14.

Corollary 3.4.11. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let $\mathcal{M}_{\mathrm{C}} \in \{0,1\}^{t \times N}$ be a (K, α)-coherent matrix, where $K > 3k\alpha$. Then for each $n \in [N]$, more than $K/3$ rows of ($\mathcal{M}_{\mathrm{C}}(n)$ and therefore) $\mathcal{M}_{\mathrm{C}}$ satisfy (3.15) with respect to n and k.

Proof. Let $n \in [N]$. Apply Lemma 3.4.10 with $c = 3$ to show that there are more than $K/3$ values of $\ell \in [t]$ for which

$$\left(\mathcal{M}_{\mathrm{C}}\right)_{\ell,n} = 1 \quad \text{and} \quad \left| \left(\mathcal{M}_{\mathrm{C}}\, |\mathbf{x}|\right)_\ell - |x_n| \right| \leq \frac{1}{k}\, \sigma_k(\mathbf{x})_1.$$

The rows of $\mathcal{M}_{\mathrm{C}}$ indexed by these values of ℓ each satisfy (3.15) with respect to n and k. □

We now formalize Remark 3.4.9 into Theorem 3.4.12, in a similar manner to the formalization of Remark 3.4.7 into Theorem 3.4.8.

Theorem 3.4.12. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix. Let $\mathcal{M}_{\mathrm{C}} = (\mathbf{r}_\ell) \in \{0,1\}^{t \times N}$ be a (K, α)-coherent matrix, where $K > 3k\alpha$. For $\ell \in [t]$, let

$$\mathbf{z}^{(\ell)} = \left(z_i^{(\ell)}\right) = \left(\mathbf{r}_\ell \circledast \mathcal{B}_N\right) \mathbf{x} \in \mathbb{R}^{1 + \lceil \log_2 N \rceil}$$

and compute

$$n_\ell = \sum_{i=1}^{\lceil \log_2 N \rceil} I\left[\left| z_i^{(\ell)} \right| > \left| z_0^{(\ell)} - z_i^{(\ell)} \right| \right] 2^{\lceil \log_2 N \rceil - i}. \tag{3.18}$$

Then the multiset $U = [n_0, n_1, \ldots, n_{t-1}]$ contains more than $K/3$ occurrences of each heavy hitter of $\mathbf{x}$ with respect to k.

Proof. Let n be a heavy hitter of $\mathbf{x}$ with respect to k. By Corollary 3.4.11, there are more than $K/3$ values of $\ell \in [t]$ for which $\mathbf{r}_\ell$ satisfies (3.15) with respect to n and k. For each such

ℓ, application of Theorem 3.4.8 with $\mathbf{r} = \mathbf{r}_\ell$ and $\mathbf{z} = \mathbf{z}^{(\ell)}$ shows that $n_\ell = n$. Therefore, U contains more than $K/3$ occurrences of n with respect to k. □

Remark 3.4.13. We shall use the identification matrix $\mathcal{M}_{\mathrm{id}} = \mathcal{M}_{\mathrm{C}} \circledast \mathcal{B}_N$. By Remark 3.4.4, the ℓ^{th} block of $\mathbf{y}_{\mathrm{id}} = \mathcal{M}_{\mathrm{id}}\,\mathbf{x}$ is then $\mathbf{z}^{(\ell)}$, so the multiset U of indices determined from the $\mathbf{z}^{(\ell)}$ in this way according to (3.18) is guaranteed to contain more than $K/3$ occurrences of each heavy hitter. Note the indices of $\mathbf{z}^{(\ell)}$ are related to those of $\mathbf{y}_{\mathrm{id}}$ by $z_i^{(\ell)} = y_{\ell(1+\lceil \log_2 N\rceil)+i}$, from (3.12).

Remark 3.4.14. Remark 3.4.13 also implies that the indices occurring at most $K/3$ times in the multiset U are certainly not heavy hitters (that is, they are false positives) and can be safely discarded. This gives us an identification algorithm (see Algorithm 7) to determine from $\mathbf{y}_{\mathrm{id}} = \mathcal{M}_{\mathrm{id}}\,\mathbf{x}$ a relatively small index set $S \subseteq [N]$ containing the heavy hitters: first form the multiset U as in Theorem 3.4.12 (see Lines 2–12 of Algorithm 7), then retain only the indices which occur more than $K/3$ times (see Line 13 of Algorithm 7). A consequence of this removal of false positives is that the size of the index set $S \subseteq [N]$ output by Algorithm 7 is less than $\frac{1}{K/3}|U| = O(t/K)$. This is why we took the value of c from Lemma 3.4.10 to be greater than 2 in deriving Corollary 3.4.11: we want $\mathcal{M}_{\mathrm{C}}$ to have $\Omega(K)$ good rows so that the size of S can be reduced from t to $O(t/K)$.

Algorithm 7 Heavy Hitter Identification for Theorem 3.4.15 [BIS12, Algorithm 1]

```
    Input: $\mathcal{M}_{\mathrm{id}} = \mathcal{R} \circledast \mathcal{B}_N \in \{0,1\}^{t(1+\lceil \log_2 N\rceil)\times N}$ and $\mathbf{y}_{\mathrm{id}} = \mathcal{M}_{\mathrm{id}}\,\mathbf{x} = \left(y_{\ell(1+\lceil \log_2 N\rceil)+i}\right)$
    Output: set $S \subseteq [N]$
 1: Initialize multiset $U \leftarrow \emptyset$
 2: for $\ell$ from 0 to $t-1$ do          ▷ determine an index $n$ from the $\ell^{\text{th}}$ block of $\mathbf{y}_{\mathrm{id}}$
 3:     for $i$ from 1 to $\lceil \log_2 N\rceil$ do
 4:         if $\left|y_{\ell(1+\lceil \log_2 N\rceil)+i}\right| > \left|y_{\ell(1+\lceil \log_2 N\rceil)} - y_{\ell(1+\lceil \log_2 N\rceil)+i}\right|$ then
 5:             $v_i \leftarrow 1$
 6:         else
 7:             $v_i \leftarrow 0$
 8:         end if
 9:     end for
10:     $n \leftarrow \sum_{i=1}^{\lceil \log_2 N\rceil} v_i\, 2^{\lceil \log_2 N\rceil - i}$
11:     $U \leftarrow U \uplus \{n\}$
12: end for
13: $K \leftarrow$ the number of 1 entries in the first column of $\mathcal{M}_{\mathrm{id}}$
14: $S \leftarrow \{n \mid n \in U$ with multiplicity greater than $K/3\}$     ▷ Remove known false positives
15: Output: $S$
```

We now describe the performance of the identification scheme $(\mathcal{M}_{\text{id}}, \Delta_{\text{id}})$ where $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$ and $\mathcal{R} = \mathcal{M}_{\text{C}}$ is a (K, α)-coherent matrix and Δ_{id} is Algorithm 7.

Theorem 3.4.15. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix. Let $\mathcal{R} = \mathcal{M}_{\text{C}} \in \{0,1\}^{t \times N}$ be a (K, α)-coherent matrix, where $K > 3k\alpha$. Let $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$ be the identification matrix and $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\, \mathbf{x}$ be the identification measurement. Then applying Algorithm 7 to inputs $\mathcal{M}_{\text{id}}$ and $\mathbf{y}_{\text{id}}$ produces an output $S \subseteq [N]$ containing every heavy hitter of $\mathbf{x}$ with respect to k, and its runtime is $\Theta(t \log N + t \log t)$. Furthermore, the row count of the identification matrix $\mathcal{M}_{\text{id}}$ is $\Theta(t \log N)$.

Proof. We first establish the correctness of the algorithm. K can be determined from $\mathcal{M}_{\text{id}}$ by counting the number of 1 entries in the first column of $\mathcal{M}_{\text{id}}$ (see Definitions 2.4.1, 3.4.1 and 3.4.2).[1] The correctness of the rest of algorithm follows directly from Theorem 3.4.12, Remarks 3.4.13 and 3.4.14. The row count of the identification matrix $\mathcal{M}_{\text{id}}$ is given by Definitions 3.4.1 and 3.4.2.

It remains to determine the runtime. Since the runtime is $\Theta(1)$ for Lines 4–8, the runtime for Lines 3–11 is $\Theta(\log N)$. Therefore, the runtime for Lines 2–12 is $\Theta(t \log N)$. The runtime for Line 13 is $\Theta(t \log N)$, because $\mathcal{M}_{\text{id}}$ has $\Theta(t \log N)$ rows. The set S in Line 14 can be constructed by sorting the elements in U in $\Theta(|U| \log |U|) = \Theta(t \log t)$ time, followed by a linear scan of the sorted data in $\Theta(t)$ time. Therefore, the runtime is $\Theta(t \log N + t \log t)$. □

To derive a good identification scheme from Theorem 3.4.15, it remains to identify a (K, α)-coherent matrix $\mathcal{M}_{\text{C}} \in \{0,1\}^{t \times N}$ with $K > 3k\alpha$ and a small row count t. This is provided by a Porat-Rothschild (K, α)-coherent matrix from Theorem 2.4.7, taking $\alpha = \Theta\left(\log N\right)$ and $K = 3k\alpha + 1 = \Theta(k \log N)$, whose row count is $t = \Theta(K^2/\alpha) = \Theta(k^2 \log N)$. This gives us an identification scheme with the following performance.

Corollary 3.4.16. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix. Let $\mathcal{R} = \mathcal{M}_{\text{PR}}$ be a Porat-Rothschild (K, α)-coherent matrix, where $\alpha = \Theta\left(\log N\right)$ and $K = 3k\alpha + 1$. Let $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$ be the identification matrix and $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\, \mathbf{x}$ be the identification measurement. Then applying Algorithm 7 to inputs $\mathcal{M}_{\text{id}}$ and $\mathbf{y}_{\text{id}}$ produces an output $S \subseteq [N]$ containing every heavy hitter of $\mathbf{x}$ with respect to k. Both the row count of the identification matrix and the runtime of Algorithm 7 are $\Theta(k^2 \log^2 N)$.

Remark 3.4.17. The index set $S \subseteq [N]$ produced by the scheme in Corollary 3.4.16 has size $|S| = O\left(\frac{k^2 \log N}{k \log N}\right) = O(k)$ by Remark 3.4.14, which is desirable because the number of heavy hitters can be at least k by Remark 3.2.2.

[1] It is not memory-efficient to store the entire identification matrix $\mathcal{M}_{\text{id}}$ just to compute K. We can modify Algorithm 7 to have an input $\mathcal{R}$ or even K to reduce this inefficiency.

3.5 Putting It All Together

We combine the identification scheme in Corollary 3.4.16, the estimation scheme in Corollary 3.3.10 and the pruning algorithm (Algorithm 5) to produce a deterministic compressed sensing scheme. In particular, the recovery algorithm (see Algorithm 8) is the composition of Algorithms 7, 6 and 5: see also Figure 3.1. To the author's knowledge, the recovery algorithm of this scheme is faster than all other known deterministic compressed sensing schemes satisfying an ℓ_p/ℓ_q guarantee. The performance of the scheme is described in Corollary 3.5.1.

Corollary 3.5.1. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Construct an identification matrix $\mathcal{M}_{\text{id}}$ according to Corollary 3.4.16 and an estimation matrix $\mathcal{M}_{\text{est}}$ according to Corollary 3.3.10. Let $\mathcal{M} = \left[\frac{\mathcal{M}_{\text{id}}}{\mathcal{M}_{\text{est}}}\right]$ be the measurement matrix and let $\mathbf{y} = \mathcal{M}\mathbf{x}$ be the measurement. Then applying Algorithm 8 to inputs $\mathcal{M}$, $\mathbf{y}$, k produces an output $\widehat{\mathbf{x}}$ satisfying

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \frac{1 + 4\sqrt{2}}{\sqrt{k}}\,\sigma_k(\mathbf{x})_1.$$

Both the row count of the measurement matrix $\mathcal{M}$ and the runtime of Algorithm 8 are $O(k^2 \log^2 N)$.

Proof. The error-guarantee of the approximation follows from Corollaries 3.4.16, 3.3.10, and 3.2.4. The row count of $\mathcal{M}$ is given by summing the row counts in Corollaries 3.4.16 and 3.3.10. We now analyze the runtime. The runtime of the identification phase is $O(k^2 \log^2 N)$ by Corollary 3.4.16. Since the output S of the identification phase satisfies $|S| = O(k)$ by Remark 3.4.17, the runtime of the estimation phase is $O(k^2 \log_k^2 N)$ by Corollary 3.3.10. The runtime of the pruning phase is $O(|S| \log |S|) = O(k \log k)$ by Corollary 3.2.4. Summing the runtime of each phase gives the desired bound. □

Remark 3.5.2. We choose the estimation scheme of our compressed sensing scheme to be from Corollary 3.3.10 rather than from Corollary 3.3.8. That is, we take the estimation matrix $\mathcal{M}_{\text{est}}$ to be a Kautz-Singleton matrix $\mathcal{M}_{\text{KS}}$ rather than a Porat-Rothschild matrix $\mathcal{M}_{\text{PR}}$. The reason is that the strongly explicit construction of $\mathcal{M}_{\text{KS}}$ allows the estimation phase of Algorithm 8 to be carried out without storing the locations of the 1 entries of the estimation matrix (see Remark 3.3.9), giving a significantly reduced memory requirement. Although using $\mathcal{M}_{\text{PR}}$ instead of $\mathcal{M}_{\text{KS}}$ would lead to fewer measurements and faster runtime for the estimation scheme, the number of measurements and the runtime for the complete compressed sensing scheme would still be $O(k^2 \log^2 N)$ (as in Corollary 3.3.10) because the bottleneck for both lies in the identification matrix and the identification phase.

Algorithm 8 Recovery Algorithm for Corollary 3.5.1 [BIS12, Algorithm 1]

```
Input: $\mathcal{M} = \left[\begin{array}{c} \mathcal{M}_{\text{id}} \\ \hline \mathcal{M}_{\text{est}} \end{array}\right] = \left[\begin{array}{c} \mathcal{R} \circledast \mathcal{B}_N \\ \hline \mathcal{M}_{\text{est}} \end{array}\right]$ and $\mathbf{y} = (y_j) = \mathcal{M}\mathbf{x} = \left[\begin{array}{c} \mathbf{y}_{\text{id}} \\ \hline \mathbf{y}_{\text{est}} \end{array}\right]$ and $k$
Output: $\widehat{\mathbf{x}} \in \mathbb{R}^N$
Initialize multiset $U \leftarrow \emptyset$ and vector $\mathbf{z} \leftarrow \mathbf{0} \in \mathbb{R}^N$

                    IDENTIFICATION PHASE (ALGORITHM 7)

for $\ell$ from $0$ to $t-1$ do                              ▷ $t =$ the row count of $\mathcal{R}$
    for $i$ from $1$ to $\lceil \log_2 N \rceil$ do
        if $\left|y_{\ell(1+\lceil \log_2 N \rceil)+i}\right| > \left|y_{\ell(1+\lceil \log_2 N \rceil)} - y_{\ell(1+\lceil \log_2 N \rceil)+i}\right|$ then
            $v_i \leftarrow 1$
        else
            $v_i \leftarrow 0$
        end if
    end for
    $n \leftarrow \sum_{i=1}^{\lceil \log_2 N \rceil} v_i \, 2^{\lceil \log_2 N \rceil - i}$
    $U \leftarrow U \uplus \{n\}$
end for
$K \leftarrow$ the number of 1 entries in the first column of $\mathcal{M}_{\text{id}}$
$S \leftarrow \{n \mid n \in U$ with multiplicity greater than $K/3\}$

                    ESTIMATION PHASE (ALGORITHM 6)

for each $n$ in $S$ do
    $z_n \leftarrow$ median of the entries of $\mathbf{y}_{\text{est}}$ corresponding to 1 entries in column $n$ of $\mathcal{M}_{\text{est}}$
end for

                    PRUNING PHASE (ALGORITHM 5)

if $2k < |S|$ then
    Sort by magnitude the entries of $\mathbf{z}_S$ so that $|z_{n_1}| \geq |z_{n_2}| \geq \cdots \geq \left|z_{n_{|S|}}\right|$
    $\widetilde{S} \leftarrow \{n_1, \ldots, n_{2k}\}$
else
    $\widetilde{S} \leftarrow S$
end if
Output: $\widehat{\mathbf{x}} = \mathbf{z}_{\widetilde{S}}$
```

Chapter 4

Compressed Sensing Scheme with a Near-Optimal Runtime and Low Randomness

We present a compressed sensing scheme (Corollary 4.4.1) with a nonuniform ℓ_2/ℓ_1 error guarantee. Both the row count and the recovery algorithm runtime are $O(k \log k \cdot \log N)$. For the regime $k \leq N^c$ for some fixed $c \in [0, 1)$, these growth rates are within a factor of $O(\log k)$ of the respective lower bounds of $\Omega(k \log(N/k))$ (see Section 2.1.1). Furthermore, our scheme is as fast as that of [Iwe14, Theorem 5 (3)], which has the fastest known runtime of schemes with an ℓ_p/ℓ_q error-guarantee (1.1), yet the entropy is reduced from $O\left(Nk^2 \log N\right)$ to $O\left(\log k \cdot \log\left(k \log N\right)\right)$. Furthermore, the entropy required is as low as that of [Iwe14, Theorem 5 (2)], which is the only previously known scheme satisfying (P1)–(P3) and requiring sublinear entropy. In Corollary 4.3.7, we provide a variant of Corollary 4.4.1, which is the first measurement-optimal scheme with sublinear entropy.

4.1 Overview of Techniques

In Chapter 3, (K, α)-coherent matrices are used to construct both the identification matrix (see Theorem 3.4.15) and the estimation matrix (see Theorem 3.3.7). In this chapter, we randomize the construction of the identification and estimation matrix, replacing each of the (K, α)-coherent matrices by a random selection of its rows. This carries two advantages: the number of measurements is reduced, and the recovery algorithm is faster. The disadvantage is that there is a small (although controllable) probability their output is incorrect. Combining the improvements of the identification and the estimation schemes, we obtain a nonuniform ℓ_2/ℓ_1 compressed sensing scheme (see Corollary 4.4.1) in which both the number of measurements and the recovery algorithm runtime are $O(k \log k \cdot \log N)$. Although we modify both the identification matrix and the estimation matrix, no change is required to the estimation and pruning algorithms of Chapter 3: only the identification algorithm need be modified (see Figure 4.1).

4.1.1 Comparison with Iwen's schemes

The compressed sensing scheme that we will present in Corollary 4.4.1 combines the advantages of two schemes S_1, S_2 proposed by Iwen [Iwe14, Theorem 5 (3) and Theorem 5 (2)]: see Table 1.1 for a summary of their performance.

Both schemes S_1, S_2 use the recovery algorithm Δ given in Algorithm 10. Their measurement matrix takes the form $\mathcal{M} = \left[\begin{array}{c}\mathcal{M}_{\text{id}} \\ \hline \mathcal{M}_{\text{est}}\end{array}\right]$, where $\mathcal{M}_{\text{id}}$ is an identification matrix and $\mathcal{M}_{\text{est}}$ is an estimation matrix. The estimation matrix $\mathcal{M}_{\text{est}}$ is obtained by randomly subsampling (with replacement) blocks of rows from a Kautz-Singleton (K, α)-coherent matrix $\mathcal{M}_{\text{KS}}$. The identification matrix $\mathcal{M}_{\text{id}}$ is the columnwise Kronecker product of the N^{th} bit-test matrix with a binary matrix obtained by randomly subsampling (with replacement) rows from some (K, α)-coherent matrix $\mathcal{M}_{\text{C}}$.

In scheme S_2, the matrix $\mathcal{M}_{\text{C}}$ is fixed to be $\mathcal{M}_{\text{KS}}$, and the required entropy is reduced by subsampling on row blocks rather than on individual rows. In contrast, in scheme S_1, the matrix $\mathcal{M}_{\text{C}}$ is itself randomly generated [Iwe14, Theorem 2], and random subsampling is then carried out on individual rows. The row count of this randomly generated matrix $\mathcal{M}_{\text{C}}$ meets the lower bound in Theorem 2.4.4, and in comparison with S_1 requires fewer measurements and gives faster runtime, but the process of random subsampling on individual rows from a randomly generated matrix requires higher entropy.

We will show in Corollary 4.4.1 that, by instead fixing $\mathcal{M}_{\text{C}}$ to be a Porat-Rothschild matrix $\mathcal{M}_{\text{PR}}$ and randomly subsampling blocks of rows, we can retain the advantages of S_1 over S_2 (fewer measurements and faster runtime) without incurring the penalty of higher entropy.

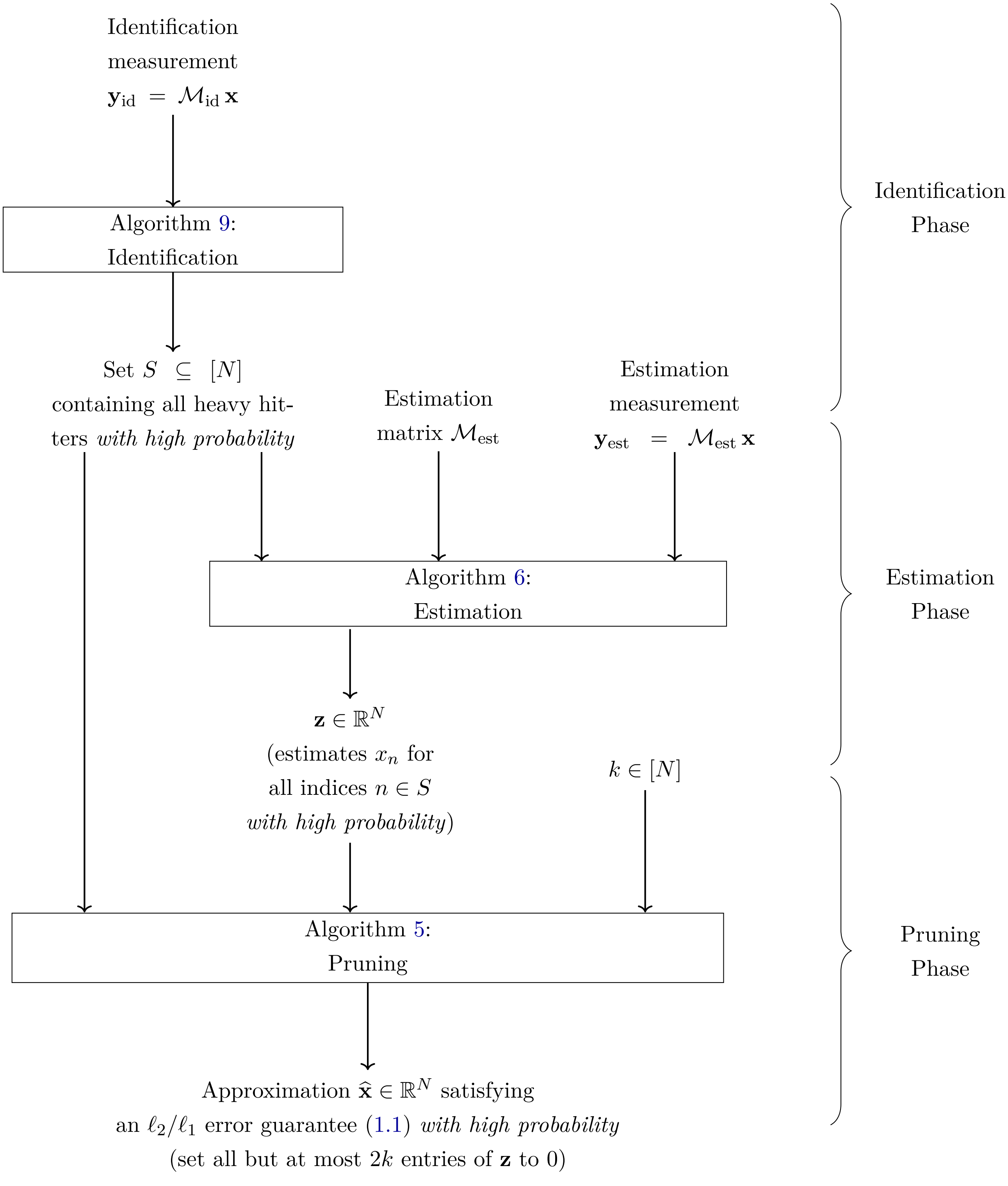

Identification measurement $\mathbf{y}_{\mathrm{id}} = \mathcal{M}_{\mathrm{id}}\,\mathbf{x}$
Algorithm 9: Identification
Set $S \subseteq [N]$ containing all heavy hitters *with high probability*
Estimation matrix $\mathcal{M}_{\mathrm{est}}$
Estimation measurement $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$
Algorithm 6: Estimation
$\mathbf{z} \in \mathbb{R}^N$ (estimates x_n for all indices $n \in S$ *with high probability*)
$k \in [N]$
Algorithm 5: Pruning
Approximation $\widehat{\mathbf{x}} \in \mathbb{R}^N$ satisfying an ℓ_2/ℓ_1 error guarantee (1.1) *with high probability* (set all but at most $2k$ entries of $\mathbf{z}$ to 0)
Identification Phase
Estimation Phase
Pruning Phase

Figure 4.1: Overview of the identification, estimation, and pruning phases of the recovery algorithm (Algorithm 10) for the randomized compressed sensing scheme of Corollary 4.4.1. This differs from the recovery algorithm shown in Figure 3.1 only in that the identification phase (Algorithm 7) is replaced by Algorithm 9 (see Remark 4.2.4).

4.2 Randomized Identification Scheme

In this section, we describe the identification phase depicted in Figure 4.1. Recall that we previously used the identification scheme of Theorem 3.4.15 to identify the heavy hitters of a vector $\mathbf{x} \in \mathbb{R}^N$ (with respect to k), namely the indices $n \in [N]$ satisfying (3.11):

$$|x_n| > \frac{1}{k}\,\sigma_k(\mathbf{x})_1.$$

This scheme relies on an identification matrix $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$, where $\mathcal{R}$ is a (K, α)-coherent matrix having t rows. Since the number of measurements and recovery algorithm runtime of the identification scheme both grow with t, we wish t to be small. The underlying property of $\mathcal{R}$ used to derive Theorem 3.4.15 is that, for each heavy hitter n, there is a “good” row $\mathbf{r} \in \{0,1\}^{1\times N}$ of $\mathcal{R}$ satisfying (3.15) with respect to n and k (see Remark 3.4.9):

$$r_n = 1 \quad \text{and} \quad \left|\sum_j r_j|x_j| - |x_n|\right| < \frac{1}{k}\,\sigma_k(\mathbf{x})_1.$$

Since the number of heavy hitters of $\mathbf{x}$ is $O(k)$ by Lemma 3.2.1, ideally the matrix $\mathcal{R}$ would have only $t = O(k)$ rows. However, Theorem 3.4.15 assumes $K > 3k\alpha$, so that the number of rows of $\mathcal{R}$ in the corresponding scheme is $t = \Omega(K^2/\alpha) = \Omega(k^2\alpha)$ by Theorem 2.4.4.

In this section, we show how to reduce the gap between $O(k)$ and $\Omega(k^2\alpha)$ by instead choosing $\mathcal{R}$ to comprise $O(k \log k)$ randomly selected rows of a Porat-Rothschild (K, α)-coherent matrix (see Corollary 4.2.6). This improvement comes at the cost of randomizing the construction of the identification matrix, with the consequence that the output of the identification scheme could be incorrect with a small (although controllable) probability.

We define

event G: for each heavy hitter n of $\mathbf{x}$ with respect to k, there is at least one row of $\mathcal{R}$ satisfying (3.15) with respect to n and k. (4.1)

We shall establish in Lemma 4.2.1 an upper bound on the number of rows of a matrix $\mathcal{R}$, each of which is randomly selected from a (K, α)-coherent matrix in block form, so that event G occurs with high probability. We shall then apply Lemma 4.2.1 to a Porat-Rothschild matrix $\mathcal{M}_{\text{PR}}$ in Corollary 4.2.6 to obtain a randomized identification scheme whose number of measurements and recovery algorithm runtime are both $O(k \log k \cdot \log N)$. Furthermore, the entropy required is only $O\left(\log k \cdot \log\left(k \log N\right)\right)$.

The following lemma is a randomized variant of Corollary 3.4.11. Note that Corollary 3.4.11 does not require n to be a heavy hitter, whereas the following lemma does. Observe that the random subsampling is done on row *blocks* rather than on individual rows, because we will apply it to $\mathcal{M}_{\mathrm{PR}}$, which is a (K,α)-coherent matrix in block form. This reduces the required entropy. A variant of this lemma which subsamples on individual rows can be found in [Iwe14, Corollary 1].

Lemma 4.2.1. Let $\sigma \in [0,1)$ and $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let $\mathcal{M}_{\mathrm{C}} \in \{0,1\}^{u\times N}$ be a (K,α)-coherent matrix comprising K blocks of $\frac{u}{K}$ rows, each column of each row block containing exactly one 1, where $K > 3k\alpha$. Construct a matrix $\mathcal{R}$ by choosing $\left\lceil \frac{1}{\ln 1.5} \ln\left(\frac{2k}{1-\sigma}\right)\right\rceil$ of the K row blocks of $\mathcal{M}_{\mathrm{C}}$ uniformly at random with replacement. Then event G (with respect to $\mathbf{x}$, k, $\mathcal{R}$ as in (4.1)) occurs with probability at least σ. Furthermore, the number of rows of $\mathcal{R}$ is $\Theta\left(\frac{u}{K}\log\left(\frac{2k}{1-\sigma}\right)\right)$ and the entropy of $\mathcal{R}$ is $O\left(\log\left(\frac{2k}{1-\sigma}\right)\cdot \log K\right)$.

Proof. The number of rows of $\mathcal{R}$ follows from the construction. We now analyze the entropy of $\mathcal{R}$. This is equivalent to the entropy of the vector-valued random variable $X = \left(X_1, X_2, \ldots, X_{\left\lceil \frac{1}{\ln 1.5}\ln\left(\frac{2k}{1-\sigma}\right)\right\rceil}\right)$, where each X_i is the outcome of drawing from the discrete uniform distribution over $[K]$. Since the X_i are independent and identically distributed random variables, it follows from [CT06b, Theorems 2.6.6 and 2.6.4] that

$$H(X) = \left\lceil \frac{1}{\ln 1.5}\ln\left(\frac{2k}{1-\sigma}\right)\right\rceil H(X_1) = \left\lceil \frac{1}{\ln 1.5}\ln\left(\frac{2k}{1-\sigma}\right)\right\rceil \log_2 K.$$

It remains to show that $\mathbb{P}(G) \geq \sigma$.

Let T be the set of all heavy hitters n of $\mathbf{x}$ with respect to k, as in (3.7). By Lemma 3.2.1, we have $|T| \leq 2k$. For $n \in T$, let G_n be the event that there is at least one row of $\mathcal{R}$ satisfying (3.15) with respect to n and k. We claim that

$$\mathbb{P}\left(\overline{G_n}\right) \leq \frac{1-\sigma}{2k} \quad \text{for each } n \in T.$$

Then

$$\begin{aligned}\mathbb{P}(G) &= \mathbb{P}\left(\bigcap_{n\in T} G_n\right)\\ &= 1 - \mathbb{P}\left(\bigcup_{n\in T}\overline{G_n}\right)\\ &\geq 1 - \sum_{n\in T}\mathbb{P}\left(\overline{G_n}\right)\\ &\geq 1 - \frac{|T|}{2k}(1-\sigma)\\ &\geq \sigma,\end{aligned}$$

where the first inequality holds by the union bound.

It remains to prove the claim. Fix $n \in T$. We require an upper bound on the probability of $\overline{G_n}$, the event that each row of $\mathcal{R}$ does not satisfy (3.15) (with respect to n and k). Equivalently, $\overline{G_n}$ is the event that each row block of $\mathcal{R}$ does not contain a row satisfying (3.15).

Now each row block of $\mathcal{M}_{\mathrm{C}}$ contains a unique row whose entry in column n is 1, and all other rows in the block fail the first condition of (3.15). Therefore, at most one row in each block can satisfy (3.15). By Corollary 3.4.11, more than $K/3$ rows of $\mathcal{M}_{\mathrm{C}}$ satisfy (3.15). It follows that more than $K/3$ of the K blocks contain a row satisfying (3.15). Therefore, the probability that a randomly chosen block does not contain a row satisfying (3.15) is less than 2/3. Therefore, applying the product rule over all $\left\lceil \frac{1}{\ln 1.5} \ln\left(\frac{2k}{1-\sigma}\right)\right\rceil$ randomly chosen blocks of $\mathcal{R}$, we conclude that

$$\mathbb{P}\left(\overline{G_n}\right) < \left(\frac{2}{3}\right)^{\left\lceil \frac{1}{\ln 1.5} \ln\left(\frac{2k}{1-\sigma}\right)\right\rceil} \leq \frac{1}{1.5^{\frac{1}{\ln 1.5} \ln\left(\frac{2k}{1-\sigma}\right)}} = \frac{1}{\exp\left(\ln\left(\frac{2k}{1-\sigma}\right)\right)} = \frac{1-\sigma}{2k}. \qquad \square$$

Remark 4.2.2. For the matrix $\mathcal{R}$ constructed in Lemma 4.2.1, the probability that event G occurs is at least σ. If G occurs, then by Remark 3.4.7 we can determine every heavy hitter of $\mathbf{x}$ with respect to k. Therefore, we can determine every heavy hitter of $\mathbf{x}$ with probability at least σ. We show explicitly how to achieve this in Theorem 4.2.3 below. This gives a randomized variant of the identification scheme in Theorem 3.4.15, as described in Theorem 4.2.5.

The following result is a randomized variant of Theorem 3.4.12.

Theorem 4.2.3. Let $\sigma \in [0,1)$ and $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix. Construct a matrix $\mathcal{R} = (\mathbf{r}_\ell) \in \{0,1\}^{t \times N}$ according to Lemma 4.2.1. For $\ell \in [t]$, let

$$\mathbf{z}^{(\ell)} = \left(z_i^{(\ell)}\right) = (\mathbf{r}_\ell \circledast \mathcal{B}_N)\,\mathbf{x} \in \mathbb{R}^{1+\lceil \log_2 N \rceil}$$

and compute (as in (3.18))

$$n_\ell = \sum_{i=1}^{\lceil \log_2 N \rceil} I\left[\left|z_i^{(\ell)}\right| > \left|z_0^{(\ell)} - z_i^{(\ell)}\right|\right] 2^{\lceil \log_2 N \rceil - i}.$$

Then with probability at least σ, the set $S = \{n_0, n_1, \ldots, n_{t-1}\}$ contains every heavy hitter of $\mathbf{x}$ with respect to k.

Proof. By Lemma 4.2.1, the event G (with respect to $\mathbf{x}$, k, $\mathcal{R}$) occurs with probability at least σ. It is therefore sufficient to assume that G occurs, and then show that S contains every heavy hitter of $\mathbf{x}$ with respect to k.

Let n be a heavy hitter of $\mathbf{x}$ with respect to k. Since event G occurs, there is at least one row $\mathbf{r}_\ell$ of $\mathcal{R}$ satisfying (3.15) with respect to n and k. Apply Theorem 3.4.8 with $\mathbf{r} = \mathbf{r}_\ell$ and $\mathbf{z} = \mathbf{z}^{(\ell)}$ to show that $n = n_\ell$. Therefore $n \in S$, as required. □

Remark 4.2.4. This provides a randomized identification scheme with a randomly constructed identification matrix $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$, which produces an identification measurement $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\,\mathbf{x}$. By relating the indices of $\mathbf{z}^{(\ell)}$ to those of $\mathbf{y}_{\text{id}}$ by $z_i^{(\ell)} = y_{\ell(1+\lceil \log_2 N \rceil)+i}$ (see (3.12)), we obtain a deterministic[1] identification algorithm (see Algorithm 9) which inputs $\mathbf{y}_{\text{id}}$ and outputs a set $S \subseteq [N]$ containing every heavy hitter. Algorithm 9 is similar to Algorithm 7, the identification algorithm of the scheme in Theorem 3.4.15. The main difference is that Algorithm 9 returns a *set* S comprising the indices determined from (3.18) whereas Algorithm 7 first calculates a *multiset* U comprising the indices determined from (3.18) and then returns a set S comprising indices whose multiplicity in U is greater than $K/3$. As the identification matrix $\mathcal{M}_{\text{id}}$ is used in Algorithm 7 only to determine K (see Line 13 of Algorithm 7), it is not a required input for Algorithm 9.

Algorithm 9 Heavy Hitter Identification for Theorem 4.2.5 [Iwe14, Algorithm 1]

```
Input: $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\,\mathbf{x} = (\mathcal{R} \circledast \mathcal{B}_N)\,\mathbf{x} = \left(y_{\ell(1+\lceil \log_2 N \rceil)+i}\right)$
Output: set $S \subseteq [N]$
Initialize set $S \leftarrow \emptyset$
for $\ell$ from $0$ to $t-1$ do                                   ▷ $t$ = the row count of $\mathcal{R}$
    for $i$ from $1$ to $\lceil \log_2 N \rceil$ do
        if $\left|y_{\ell(1+\lceil \log_2 N \rceil)+i}\right| > \left|y_{\ell(1+\lceil \log_2 N \rceil)} - y_{\ell(1+\lceil \log_2 N \rceil)+i}\right|$ then
            $v_i \leftarrow 1$
        else
            $v_i \leftarrow 0$
        end if
    end for
    $n \leftarrow \sum_{i=1}^{\lceil \log_2 N \rceil} v_i\, 2^{\lceil \log_2 N \rceil - i}$
    $S \leftarrow S \cup \{n\}$
end for
Output: $S$
```

Theorem 4.2.5. Let $\sigma \in [0,1)$ and $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix. Construct a matrix $\mathcal{R} = (\mathbf{r}_\ell) \in \{0,1\}^{t \times N}$ according to Lemma 4.2.1, and let u be the row count of the associated (K, α)-coherent matrix $\mathcal{M}_{\text{C}}$. Let $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$

[1] although the identification scheme (matrix/algorithm pair) is randomized and has a nonzero probability of returning incorrect output, the recovery algorithm itself is deterministic. In particular, it will *always* return a set $S \subseteq [N]$ containing every heavy hitter if event G occurs.

be the identification matrix and $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\,\mathbf{x}$ be the identification measurement. Then applying Algorithm 9 to input $\mathbf{y}_{\text{id}}$ produces an output $S \subseteq [N]$ which, with probability at least σ, contains every heavy hitter of $\mathbf{x}$ with respect to k. Furthermore, both the runtime of Algorithm 9 and the row count of $\mathcal{M}_{\text{id}}$ are $\Theta(t \log N) = \Theta\left(\frac{u}{K} \log\left(\frac{2k}{1-\sigma}\right) \cdot \log N\right)$, and the entropy of $\mathcal{M}_{\text{id}}$ is $O\left(\log\left(\frac{2k}{1-\sigma}\right) \cdot \log K\right)$.

Proof. The required property of the set S follows directly from Theorem 4.2.3 and Remark 4.2.4. The entropy of $\mathcal{M}_{\text{id}}$ is the same as the entropy of $\mathcal{R}$, which is $O\left(\log\left(\frac{2k}{1-\sigma}\right) \cdot \log K\right)$ by Lemma 4.2.1. The row count t of $\mathcal{R}$ is $\Theta\left(\frac{u}{K} \log\left(\frac{2k}{1-\sigma}\right)\right)$, by Lemma 4.2.1. The row count of the identification matrix $\mathcal{M}_{\text{id}}$ then follows from Definitions 3.4.1 and 3.4.2. The analysis of the runtime is similar to that given in the proof of Theorem 3.4.15 for Algorithm 7: since the runtime is $\Theta(1)$ for Lines 4–8, the runtime for Lines 3–11 is $\Theta(\log N)$ and the runtime for Lines 2–12 is $\Theta(t \log N) = \Theta\left(\frac{u}{K} \log\left(\frac{2k}{1-\sigma}\right) \cdot \log N\right)$. □

To derive a good (randomized) identification scheme from Theorem 4.2.5, it remains to identify a (K, α)-coherent matrix $\mathcal{M}_{\text{C}} \in \{0,1\}^{s \times N}$ in block form with $K > 3k\alpha$ and a small row count u. This is provided by a Porat-Rothschild (K, α)-coherent matrix from Theorem 2.4.7, taking $\alpha = \Theta(\log N)$ and $K = 3k\alpha + 1 = \Theta(k \log N)$, and therefore having a row count $u = \Theta(K^2/\alpha) = \Theta(k^2 \log N)$. In particular, the row count of the corresponding matrix $\mathcal{R}$ satisfies $t = \Theta\left(\frac{u}{K} \log\left(\frac{2k}{1-\sigma}\right)\right) = \Theta\left(k \log\left(\frac{2k}{1-\sigma}\right)\right)$. By taking $\sigma = 0.99$, we obtain an identification scheme with the following performance. This is a randomized variant of Corollary 3.4.16.

Corollary 4.2.6. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Let $\mathcal{B}_N$ be the N^{th} bit-test matrix. Let $\mathcal{M}_{\text{PR}}$ be a Porat-Rothschild (K, α)-coherent matrix, where $\alpha = \Theta(\log N)$ and $K = 3k\alpha + 1$. Construct a matrix $\mathcal{R}$ by choosing $\left\lceil \frac{1}{\ln 1.5} \ln(200k) \right\rceil$ of the K row blocks of $\mathcal{M}_{\text{PR}}$ uniformly at random with replacement. Let $\mathcal{M}_{\text{id}} = \mathcal{R} \circledast \mathcal{B}_N$ be the identification matrix and $\mathbf{y}_{\text{id}} = \mathcal{M}_{\text{id}}\,\mathbf{x}$ be the identification measurement. Then applying Algorithm 9 to input $\mathbf{y}_{\text{id}}$ produces an output $S \subseteq [N]$ which, with probability at least 0.99, contains every heavy hitter of $\mathbf{x}$ with respect to k. Furthermore, both the runtime of Algorithm 9 and the row count of $\mathcal{M}_{\text{id}}$ are $\Theta(k \log k \cdot \log N)$, and the entropy of $\mathcal{M}_{\text{id}}$ is $O(\log k \cdot \log(k \log N))$.

Remark 4.2.7. The row count of $\mathcal{R}$ used in the scheme in Corollary 4.2.6 is $t = \Theta(k \log k)$, which is a factor of $O(\log k)$ from being optimal (see the discussion at the start of this section). Furthermore, the set S in Corollary 4.2.6 satisfies $|S| = t = \Theta(k \log k)$ (see Lines 2–12 of Algorithm 9). In comparison, the row count of $\mathcal{R}$ in the scheme in Corollary 3.4.16 is $t = \Theta(k^2 \log N)$, and the size of the output set S is $O(k)$ (see Remark 3.4.17).

4.3 Randomized Estimation Scheme

In this section, we describe the estimation phase depicted in Figure 4.1. Recall that we previously used the estimation scheme of Theorem 3.3.7 to produce an output $\mathbf{z} = (z_n) \in \mathbb{R}^N$ which estimates the entries of $\mathbf{x} = (x_n) \in \mathbb{R}^N$ for values of n in an arbitrary index set $S \subseteq [N]$ with high accuracy as in (3.3), namely

$$|x_n - z_n| \leq \frac{1}{k}\,\sigma_k(\mathbf{x})_1 \quad \text{for each } n \in S.$$

In Section 4.4, we shall take S to be the output of the identification scheme in Corollary 4.2.6, so that $|S| = O(k \log k)$.

The estimation scheme of Theorem 3.3.7 relies on an estimation matrix $\mathcal{M}_{\text{est}}$ having t rows, where each column of $\mathcal{M}_{\text{est}}$ contains exactly w 1s. Since the number of estimation measurements is t and the recovery algorithm runtime is $\Theta\left(|S|w\right)$, we wish both t and w to be small. In Theorem 3.3.7, we took $\mathcal{M}_{\text{est}}$ to be a (K, α)-coherent matrix, so that $w = K$ by Definition 2.4.1. Since we required that $K > 4k\alpha$, it follows that the number of rows of $\mathcal{M}_{\text{est}}$ in the corresponding scheme has $t = \Omega(K^2/\alpha) = \Omega(k^2\alpha)$ by Theorem 2.4.4, and the recovery algorithm runtime is $\Omega(|S|w) = \Omega(|S|K) = \Omega(k^2\alpha \log k)$.

We shall instead provide two estimation matrices $\mathcal{M}_{\text{est}}$, each comprising randomly selected row blocks of a (K, α)-coherent matrix: the first one from a Porat-Rothschild matrix $\mathcal{M}_{\text{PR}}$ (see Corollary 4.3.4), and the second one from a Kautz-Singleton matrix $\mathcal{M}_{\text{KS}}$ (see Corollary 4.3.6). Both estimations have $t = O(k \log |S| \cdot \log_k N)$ and $w = O(\log |S|)$, and the resulting compressed sensing scheme[2] has a performance of $O(k \log k \cdot \log N)$ for both measurements and runtime: see Section 4.4. The improvement comes at the cost of randomizing the construction of the estimation matrix, with the consequence that the output of the estimation scheme could be incorrect with a small (although controllable) probability.

The underlying property of $\mathcal{M}_{\text{est}}$ used to derive Theorem 3.3.7 is that, for each index $n \in S$, the median of the w entries of $\mathcal{M}_{\text{est}}(n)\,\mathbf{x}$ estimates x_n to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$ (see Corollary 3.3.5). We shall establish in Theorem 4.3.1 an upper bound on the number of rows of an estimation matrix $\mathcal{M}_{\text{est}}$, each of which is randomly selected from a (K, α)-coherent matrix in block form, so that the following event occurs with high probability. We

[2] comprising the identification scheme described in Corollary 4.2.6, the estimation scheme of this section, and the pruning algorithm described in Corollary 3.2.4

define

$$\text{event } E\text{: for each index } n \in S\text{, the median of the entries of the vector } \mathcal{M}_{\text{est}}(n)\,\mathbf{x} \text{ approximates } x_n \text{ to within } \tfrac{1}{k}\,\sigma_k(\mathbf{x})_1. \tag{4.2}$$

We then apply Theorem 4.3.1 to obtain Theorem 4.3.3, from which Corollaries 4.3.4 and 4.3.6 are derived.

Theorem 4.3.1 can be viewed as a randomized variant of Corollary 3.3.5. However, there is a significant difference: whereas the row count of the estimation matrix in Corollary 3.3.5 is independent of $|S|$, that of Theorem 4.3.1 grows (logarithmically) with respect to $|S|$. The random selection of rows in Theorem 4.3.1 is carried out on row *blocks* rather than on individual rows, so that by taking advantage of the block form of $\mathcal{M}_{\text{PR}}$ in Corollary 4.3.4 and $\mathcal{M}_{\text{KS}}$ in Corollary 4.3.6, we can reduce the required entropy. A version of Theorem 4.3.1 involving random selection of individual rows is given in [Iwe14, Corollary 2].

Theorem 4.3.1. Let $\sigma \in [0,1)$ and $\mathbf{x} \in \mathbb{R}^N$, let $k \in [N]$ be nonzero, and let $1 \le s \le N$. Let $\mathcal{M}_{\text{C}} \in \{0,1\}^{u \times N}$ be a (K,α)-coherent matrix comprising K blocks of $\frac{u}{K}$ rows, each column of each row block containing exactly one 1, where $K > 14k\alpha$. Construct an estimation matrix $\mathcal{M}_{\text{est}}$ by choosing $\beta := \left\lceil \frac{336}{25} \ln\left(\frac{s}{1-\sigma}\right)\right\rceil$ of the K row blocks of $\mathcal{M}_{\text{C}}$ uniformly at random with replacement. Let $S \subseteq [N]$ with $|S| = s$. Then event E (with respect to $\mathbf{x}$, k, S, $\mathcal{M}_{\text{est}}$ as in (4.2)) occurs with probability at least σ. Furthermore, the number of rows of $\mathcal{M}_{\text{est}}$ is $\Theta\left(\frac{u}{K}\log\left(\frac{s}{1-\sigma}\right)\right)$, the number of 1s in each column of $\mathcal{M}_{\text{est}}$ is β, and the entropy of $\mathcal{M}_{\text{est}}$ is $O\left(\log\left(\frac{s}{1-\sigma}\right)\cdot \log K\right)$.

Proof. The number of rows of $\mathcal{M}_{\text{est}}$ and the number of 1s in each column of $\mathcal{M}_{\text{est}}$ follow directly from the construction. The bound on the entropy is similar to that given in the proof of Theorem 4.2.1. It remains to show $\mathbb{P}(E) \ge \sigma$.

For $n \in S$, let E_n be the event that the median of the entries of the vector $\mathcal{M}_{\text{est}}(n)\,\mathbf{x}$ approximates x_n to within $\frac{1}{k}\,\sigma_k(\mathbf{x})_1$. We claim that

$$\mathbb{P}\left(\overline{E_n}\right) \le \frac{1-\sigma}{s} \quad \text{for each } n \in S.$$

Then

$$\mathbb{P}(E) = \mathbb{P}\left(\bigcap_{n\in S} E_n\right) = 1 - \mathbb{P}\left(\bigcup_{n\in S} \overline{E_n}\right) \ge 1 - \sum_{n\in S} \mathbb{P}\left(\overline{E_n}\right) \ge \sigma,$$

where the first inequality holds by the union bound, and the second using the claim and the relation $|S| = s$.

It remains to prove the claim. Fix $n \in S$. The number of 1s in column n of $\mathcal{M}_{\text{est}}$ is β, so the vector $\mathcal{M}_{\text{est}}(n)\,\mathbf{x}$ has β entries. Let A_n be the event that the inequality

$$\left|(\mathcal{M}_{\text{est}}(n)\,\mathbf{x})_j - x_n\right| < \frac{1}{k}\,\sigma_k(\mathbf{x})_1 \tag{4.3}$$

holds for more than $\beta/2$ values of $j \in [\beta]$. The event A_n is a subset of event E_n, and so $\mathbb{P}\left(\overline{E_n}\right) \leq \mathbb{P}\left(\overline{A_n}\right)$. It is therefore sufficient to show $\mathbb{P}\left(\overline{A_n}\right) \leq \frac{1-\sigma}{s}$.

For each $j \in [\beta]$, let X_j be the Bernoulli random variable indicating whether (4.3) holds for j. We are required to show that

$$\mathbb{P}\left[\sum_{j=0}^{\beta-1} X_j \leq \frac{\beta}{2}\right] \leq \frac{1-\sigma}{s}.$$

We claim that

$$\mathbb{P}\left(X_j = 1\right) > \frac{6}{7}.$$

Then

$$\mu = \mathbb{E}\left[\sum_{j=0}^{\beta-1} X_j\right] = \sum_{j=0}^{\beta-1} \mathbb{E}\left[X_j\right] = \sum_{j=0}^{\beta-1} \mathbb{P}\left(X_j = 1\right) > \frac{6}{7}\beta,$$

so that

$$\begin{aligned}
\mathbb{P}\left[\sum_{j=0}^{\beta-1} X_j \leq \frac{\beta}{2}\right] &= \mathbb{P}\left[\sum_{j=0}^{\beta-1} X_j \leq \left(1 - \frac{5}{12}\right)\frac{6}{7}\beta\right] \\
&\leq \mathbb{P}\left[\sum_{j=0}^{\beta-1} X_j \leq \left(1 - \frac{5}{12}\right)\mu\right] \\
&\leq \mathrm{e}^{-\frac{\mu(5/12)^2}{2}} \\
&< \mathrm{e}^{-\frac{25}{336}\beta} \\
&\leq \frac{1-\sigma}{s},
\end{aligned}$$

where the second inequality follows from the Chernoff bound.

It remains to show $\mathbb{P}\left(X_j = 1\right) > \frac{6}{7}$. Apply Theorem 3.3.4 with $c = 14$ to $\mathcal{M}_{\text{C}}$ to show that there are more than $\frac{6}{7}K$ values of $\ell \in [K]$ for which

$$\left|\,(\mathcal{M}_{\text{C}}(n)\,\mathbf{x})_\ell - x_n\right| < \frac{1}{k}\,\sigma_k(\mathbf{x})_1.$$

Since each of the K row blocks of $\mathcal{M}_{\mathrm{C}}$ contains a unique row whose entry in column n is 1, it follows that for ℓ drawn uniformly from $[K]$, the inequality

$$\left|\langle \text{unique row of the } \ell^{\text{th}} \text{ block of } \mathcal{M}_{\mathrm{C}}, \mathbf{x}\rangle - x_n\right| = |(\mathcal{M}_{\mathrm{C}}(n)\,\mathbf{x})_\ell - x_n| < \frac{1}{k}\,\sigma_k(\mathbf{x})_1$$

holds with probability more than 6/7. It follows that for each $j \in [\beta]$, the inequality

$$\left|\langle \text{unique row of the } j^{\text{th}} \text{ block of } \mathcal{M}_{\mathrm{est}}, \mathbf{x}\rangle - x_n\right| = \left|(\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x})_j - x_n\right| < \frac{1}{k}\,\sigma_k(\mathbf{x})_1$$

holds with probability more than 6/7. It follows that $\mathbb{P}\left(X_j = 1\right) > \frac{6}{7}$ as desired. □

Remark 4.3.2. Theorem 4.3.1 allows us to estimate entries of $\mathbf{x}$ at all indices $n \in S$ accurately (with high probability) from the vector $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$, using Algorithm 6. Furthermore, (similar to Remark 3.3.6), $\mathcal{M}_{\mathrm{est}}(n)\,\mathbf{x}$ can be computed from $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ by selecting entries according to the locations of the 1 entries in column n of $\mathcal{M}_{\mathrm{est}}$. This gives us a (randomized) estimation scheme $(\mathcal{M}_{\mathrm{est}}, \Delta_{\mathrm{est}})$, where $\mathcal{M}_{\mathrm{est}}$ is as described in Theorem 4.3.1 and Δ_{est} is Algorithm 6.

The following theorem is a randomized variant of Theorem 3.3.7.

Theorem 4.3.3. Let $\sigma \in [0, 1)$ and $\mathbf{x} \in \mathbb{R}^N$, let $k \in [N]$ be nonzero, and let $1 \leq s \leq N$. Construct an estimation matrix $\mathcal{M}_{\mathrm{est}}$ according to Theorem 4.3.1, and let u be the row count of the associated (K, α)-coherent matrix $\mathcal{M}_{\mathrm{C}}$. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ be the estimation measurement, and let $S \subseteq [N]$ with $|S| = s$. Then applying Algorithm 6 to inputs S, $\mathcal{M}_{\mathrm{est}}$, $\mathbf{y}_{\mathrm{est}}$ produces an output $\mathbf{z}$ which, with probability at least σ, satisfies (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. Assuming that the locations of the 1 entries in column n of $\mathcal{M}_{\mathrm{C}}$ are stored for each n, the runtime of Algorithm 6 is $\Theta\left(s \log\left(\frac{s}{1-\sigma}\right)\right)$. Furthermore, the row count of $\mathcal{M}_{\mathrm{est}}$ is $\Theta\left(\frac{u}{K} \log\left(\frac{s}{1-\sigma}\right)\right)$, and the entropy of $\mathcal{M}_{\mathrm{est}}$ is $O\left(\log\left(\frac{s}{1-\sigma}\right) \cdot \log K\right)$.

Proof. The row count and entropy are given by Theorem 4.3.1. We now bound the runtime. By Theorem 4.3.1, the number of 1s in each column of $\mathcal{M}_{\mathrm{est}}$ is $\beta = \Theta\left(\log\left(\frac{s}{1-\sigma}\right)\right)$. For each $n \in S$, the locations of the β entries '1' in column n of $\mathcal{M}_{\mathrm{est}}$ are stored, and so the median in Line 3 of Algorithm 6 can be computed in $\Theta(\beta)$ time using the median-of-medians algorithm [BFP+73]. Therefore, the runtime of Lines 2–5 of Algorithm 6 is $\Theta(|S|\beta) = \Theta\left(s \log\left(\frac{s}{1-\sigma}\right)\right)$.

We now show the required property of $\mathbf{z}$. It follows from Line 1 and Lines 2–5 of Algorithm 6 that $z_n = 0$ for each $n \notin S$, so $\mathbf{z}$ always satisfies (3.2) with respect to $\mathbf{x}$, k, and S. It is therefore sufficient to demonstrate that $\mathbf{z}$ satisfies (3.3) with probability at least σ. The event that $\mathbf{z}$ satisfies (3.3) is the same as the event E (with respect to $\mathbf{x}$, k, S, $\mathcal{M}_{\mathrm{est}}$ as in (4.2)) by Line 4 of Algorithm 6, and so occurs with probability at least σ by Theorem 4.3.1. □

To derive a good (randomized) estimation scheme from Theorem 4.3.3, it remains to identify a (K,α)-coherent matrix $\mathcal{M}_{\mathrm{C}} \in \{0,1\}^{u\times N}$ with $K > 14k\alpha$ and a small row count u. This can be obtained by taking a Porat-Rothschild (K,α)-coherent matrix from Theorem 2.4.7 where $\alpha = \Theta(\log N)$ and $K = 4k\alpha + 1 = \Theta(k \log N)$, whose row count is $u = \Theta(K^2/\alpha) = \Theta(k^2 \log N)$. By taking $\sigma = 0.99$, we obtain an estimation scheme with the following performance. This is a randomized variant of Corollary 3.3.8.

Corollary 4.3.4. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero, and let $1 \le s \le N$. Let $\mathcal{M}_{\mathrm{PR}}$ be a Porat-Rothschild (K,α)-coherent matrix, where $\alpha = \Theta(\log N)$ and $K = 14k\alpha + 1$. Construct an estimation matrix $\mathcal{M}_{\mathrm{est}}$ by choosing $\left\lceil \frac{336}{25} \ln(100s) \right\rceil$ of the K blocks of $\mathcal{M}_{\mathrm{PR}}$ uniformly at random with replacement. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ be the estimation measurement. Let $S \subseteq [N]$ with $|S| = s$. Then applying Algorithm 6 to inputs S, $\mathcal{M}_{\mathrm{est}}$, $\mathbf{y}_{\mathrm{est}}$ produces an output $\mathbf{z}$ which, with probability at least 0.99, satisfies (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. Assuming that the locations of the 1 entries of column n of $\mathcal{M}_{\mathrm{est}}$ are stored for each n, the runtime of Algorithm 6 is $\Theta(s \log s)$. Furthermore, the row count of $\mathcal{M}_{\mathrm{est}}$ is $\Theta(k \log s)$ and the entropy of $\mathcal{M}_{\mathrm{est}}$ is $O(\log s \cdot \log(k \log N))$.

Remark 4.3.5. We also provide a low-memory variant of Corollary 4.3.4 (see also Remark 3.3.9 for the deterministic and low-memory variant). For this estimation scheme, we use a Kautz-Singleton (K,α)-coherent matrix $\mathcal{M}_{\mathrm{KS}} \in \{0,1\}^{u\times N}$ from Theorem 2.4.6 where $\alpha = \log_k N$ and $K = 4k\alpha + 1 = \Theta(k \log_k N)$. The row count of $\mathcal{M}_{\mathrm{KS}}$ is $u = \Theta(K^2) = \Theta(k^2 \log_k^2 N)$.

Corollary 4.3.6. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero, and let $1 \le s \le N$. Let $\mathcal{M}_{\mathrm{KS}}$ be a Kautz-Singleton (K,α)-coherent matrix, where $\alpha = \log_k N$ and $K = 14k\alpha + 1$. Construct an estimation matrix $\mathcal{M}_{\mathrm{est}}$ by choosing $\left\lceil \frac{336}{25} \ln(100u) \right\rceil$ of the K blocks of $\mathcal{M}_{\mathrm{KS}}$ uniformly at random with replacement. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\,\mathbf{x}$ be the estimation measurement. Let $S \subseteq [N]$ with $|S| = s$. Then applying Algorithm 6 to inputs S, $\mathcal{M}_{\mathrm{est}}$, $\mathbf{y}_{\mathrm{est}}$ produces an output $\mathbf{z}$ which, with probability at least 0.99, satisfies (3.2) and (3.3) with respect to $\mathbf{x}$, k, and S. Assuming that the block numbers of the chosen blocks are stored, the runtime of Algorithm 6 is $O(s \log s \cdot \log_k N)$. Furthermore, the row count of $\mathcal{M}_{\mathrm{est}}$ is $\Theta(k \log_k N \log s)$, and the entropy of $\mathcal{M}_{\mathrm{est}}$ is $O(\log s \cdot \log(k \log_k N))$.

Proof. The only modification to the scheme described in Corollary 4.3.4 is that Line 3 of Algorithm 6 now determines the locations of the 1 entries of column n of $\mathcal{M}_{\mathrm{est}}$. Each of the $\left\lceil \frac{336}{25} \ln(100s) \right\rceil$ row blocks contains exactly one 1 entry in column n of $\mathcal{M}_{\mathrm{est}}$, and locating each of them takes $O(\log_K N)$ time by Theorem 2.4.6. Therefore, each instance of Line 3 takes $O(\log s \cdot \log_K N)$ time and so the runtime is $O(|S| \log s \cdot \log_K N) = O(s \log s \cdot \log_k N)$, using $K > k$. □

The (randomized) estimation schemes of Corollaries 4.3.4 and 4.3.6 allow us to estimate the entries of $\mathbf{x}$ at an arbitrary index set $S \subseteq [N]$. In particular, when the index set S contains the set T defined in (3.7), by Corollary 3.2.4 we obtain an approximation $\widehat{\mathbf{x}}$ using Algorithm 5 satisfying an ℓ_2/ℓ_1 error guarantee with probability at least 0.99. The condition that S contains T can be trivially satisfied by taking $S = [N]$ (and so $s = |S| = N$). This gives Corollary 4.3.7 (derived from Corollary 4.3.4), which describes an identification-free scheme (see Algorithm 4 and the discussion preceding it) consisting of only an estimation matrix, estimation algorithm, and pruning algorithm (Algorithm 5). This gives us a nonuniform ℓ_2/ℓ_1 compressed sensing scheme using only $O(k \log N)$ measurements, which is order-optimal for the regime $k \leq N^c$ where $c \in [0, 1)$ is arbitrary. To the author's knowledge, this scheme is the first measurement-optimal ℓ_p/ℓ_q scheme with sublinear entropy.

Corollary 4.3.7. Let $\mathbf{x} \in \mathbb{R}^N$ and let $k \in [N]$ be nonzero. Let $\mathcal{M}_{\mathrm{PR}}$ be a Porat-Rothschild (K, α)-coherent matrix, where $\alpha = \Theta(\log N)$ and $K = 14k\alpha + 1$. Construct an estimation matrix $\mathcal{M}_{\mathrm{est}}$ by choosing $\left\lceil \frac{336}{25} \ln(100N) \right\rceil$ of the K blocks of $\mathcal{M}_{\mathrm{KS}}$ uniformly at random with replacement. Let $\mathbf{y}_{\mathrm{est}} = \mathcal{M}_{\mathrm{est}}\, \mathbf{x}$ be the estimation measurement. Let $\mathbf{z}$ be the output of Algorithm 6 with inputs $S = [N]$, $\mathcal{M}_{\mathrm{est}}$, $\mathbf{y}_{\mathrm{est}}$. Then applying Algorithm 5 to inputs $\mathbf{z}$, k and $S = [N]$ produces an output $\widehat{\mathbf{x}}$ which, with probability at least 0.99, satisfies

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \frac{1 + 4\sqrt{2}}{\sqrt{k}}\, \sigma_k(\mathbf{x})_1.$$

Assuming that the locations of the 1 entries of column n of $\mathcal{M}_{\mathrm{est}}$ are stored for each n, the total runtime of Algorithms 6 and 5 is $\Theta(N \log N)$. Furthermore, the row count of $\mathcal{M}_{\mathrm{est}}$ is $\Theta(k \log N)$ and the entropy of $\mathcal{M}_{\mathrm{est}}$ is $O(\log N \cdot \log(k \log N))$.

4.4 Putting It All Together

We combine the randomized identification scheme in Corollary 4.2.6, the estimation scheme in Corollary 4.3.6 and the pruning algorithm (Algorithm 5) to produce a near-optimal nonuniform ℓ_2/ℓ_1 compressed sensing scheme (Corollary 4.4.1). This scheme is a randomized variant of Corollary 3.5.1. The recovery algorithm (see Algorithm 10) is the composition of Algorithms 9, 6 and 5: see also Figure 4.1. The recovery algorithm of this scheme is as fast (in growth rate) as that of [Iwe14, Theorem 5 (3)], which to the author's knowledge has the fastest known runtime of schemes with an ℓ_p/ℓ_q error-guarantee (1.1), yet the entropy required is reduced from $O(Nk^2 \log N)$ to $O(\log k \cdot \log(k \log N))$. The entropy required is as low as that of [Iwe14, Theorem 5 (2)], which is the only previously known scheme satisfying (P1)–(P3) and requiring sublinear entropy.

The compressed sensing scheme we seek is non-adaptive, so the estimation measurement matrix $\mathcal{M}_{\text{est}}$ must be constructed before $\mathbf{y}_{\text{id}}$ is determined. In particular, the set S output by Algorithm 9 is not known when constructing $\mathcal{M}_{\text{est}}$. However, the precise size s of the set S is known in advance via the matrix $\mathcal{R}$ (see Remark 4.2.7), and this is sufficient to construct $\mathcal{M}_{\text{est}}$ using Corollary 4.3.6. In contrast, the construction of $\mathcal{M}_{\text{est}}$ in Corollary 3.5.1 is independent of the size of the set S.

Corollary 4.4.1. Let $\mathbf{x} \in \mathbb{R}^N$, and let $k \in [N]$ be nonzero. Construct an identification matrix $\mathcal{M}_{\text{id}}$ according to Corollary 4.2.6 and let t be the row count of the matrix $\mathcal{R}$ used. Construct an estimation matrix $\mathcal{M}_{\text{est}}$ according to Corollary 4.3.6 by taking $s = t$. Let $\mathcal{M} = \left[\frac{\mathcal{M}_{\text{id}}}{\mathcal{M}_{\text{est}}}\right]$ be the measurement matrix and let $\mathbf{y} = \mathcal{M}\mathbf{x}$ be the measurement. Then applying Algorithm 10 to inputs $\mathcal{M}_{\text{est}}$, $\mathbf{y}$, k produces an output $\widehat{\mathbf{x}}$ which, with probability at least 0.99^2, satisfies

$$\|\mathbf{x} - \widehat{\mathbf{x}}\|_2 \leq \frac{1 + 4\sqrt{2}}{\sqrt{k}}\,\sigma_k(\mathbf{x})_1.$$

Both the row count of the measurement matrix $\mathcal{M}$ and (assuming that the block numbers of the chosen blocks are stored) the runtime of Algorithm 10 are $O(k \log k \cdot \log N)$. Furthermore, the entropy of $\mathcal{M}$ is $O\left(\log k \cdot \log\left(k \log N\right)\right)$.

Proof. The size of the set S output by the identification phase of Algorithm 10 is t, by Remark 4.2.7. Therefore, we may construct the estimation matrix of Corollary 4.3.6 using $s = |S| = t$. The required property of the approximation $\widehat{\mathbf{x}}$ then follows from Corollaries 4.2.6, 4.3.6, and 3.2.4.

Furthermore, $s = t = \Theta(k \log k)$ by Remark 4.2.7. It follows from Corollary 4.3.6 that the row count of $\mathcal{M}_{\text{est}}$ is $\Theta\left(k \cdot \frac{\log N}{\log k} \cdot \log(k \log k)\right) = \Theta(k \log N)$ and the entropy of $\mathcal{M}_{\text{est}}$ is $O(\log k \cdot \log(k \log_k N))$. Summing with the corresponding values from Corollary 4.2.6 gives the bound on row count and entropy for $\mathcal{M}$.

We now analyze the runtime. The runtime of the identification phase is $O(k \log k \cdot \log N)$ by Corollary 4.2.6. The runtime of the estimation phase is $O\left(s \log s \cdot \log_k N\right) = O(k \log k \cdot \log N)$ by Corollary 4.3.6. The runtime of the pruning phase is $O\left(|S| \log |S|\right) = O(k \log^2 k)$ by Corollary 3.2.4. Summing the runtime of each phase gives the desired bound. □

Algorithm 10 Recovery Algorithm for Corollary 4.4.1 [Iwe14, Algorithm 1]

```
Input: $\mathcal{M}_{\text{est}}$ and $\mathbf{y} = (y_j) = \mathcal{M}\mathbf{x} = \left[\begin{array}{c} \mathcal{M}_{\text{id}} \\ \hline \mathcal{M}_{\text{est}} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathcal{R} \circledast \mathcal{B}_N \\ \hline \mathcal{M}_{\text{est}} \end{array}\right] \mathbf{x} = \left[\begin{array}{c} \mathbf{y}_{\text{id}} \\ \hline \mathbf{y}_{\text{est}} \end{array}\right]$ and $k$
Output: $\widehat{\mathbf{x}} \in \mathbb{R}^N$
Initialize set $S \leftarrow \emptyset$ and vector $\mathbf{z} \leftarrow \mathbf{0} \in \mathbb{R}^N$

                    Identification Phase (Algorithm 9)

for $\ell$ from 0 to $t-1$ do                              ▷ $t$ = the row count of $\mathcal{R}$
    for $i$ from 1 to $\lceil \log_2 N \rceil$ do
        if $\left|y_{\ell(1+\lceil \log_2 N \rceil)+i}\right| > \left|y_{\ell(1+\lceil \log_2 N \rceil)} - y_{\ell(1+\lceil \log_2 N \rceil)+i}\right|$ then
            $v_i \leftarrow 1$
        else
            $v_i \leftarrow 0$
        end if
    end for
    $n \leftarrow \sum_{i=1}^{\lceil \log_2 N \rceil} v_i \, 2^{\lceil \log_2 N \rceil - i}$
    $S \leftarrow S \cup \{n\}$
end for

                    Estimation Phase (Algorithm 6)

for each $n$ in $S$ do
    $z_n \leftarrow$ median of the entries of $\mathbf{y}_{\text{est}}$ corresponding to 1 entries in column $n$ of $\mathcal{M}_{\text{est}}$
end for

                    Pruning Phase (Algorithm 5)

if $2k < |S|$ then
    Sort by magnitude the entries of $\mathbf{z}_S$ so that $|z_{n_1}| \geq |z_{n_2}| \geq \cdots \geq \left|z_{n_{|S|}}\right|$
    $\widetilde{S} \leftarrow \{n_1, \ldots, n_{2k}\}$
else
    $\widetilde{S} \leftarrow S$
end if
Output: $\widehat{\mathbf{x}} = \mathbf{z}_{\widetilde{S}}$
```

Chapter 5

Conclusions and Future Work

5.1 Conclusion

In this thesis, we have presented a deterministic compressed sensing scheme (Corollary 3.5.1) and a nonuniform compressed sensing scheme (Corollary 4.4.1).

The deterministic compressed sensing scheme satisfies an ℓ_2/ℓ_1 error guarantee, and both the row count and the recovery algorithm runtime are $O(k^2 \log^2 N)$. To the author's knowledge, the recovery algorithm is faster than all other deterministic compressed sensing schemes satisfying an ℓ_p/ℓ_q error guarantee (1.1). The scheme combines ideas from disjunct matrix constructions of [PR11, KS64] for the measurement matrix (see Sections 2.4.2 and 2.4.3) with a recovery algorithm using the non-iterative combinatorial approach due to [BIS12].

The nonuniform compressed sensing scheme is a randomized variant of the deterministic scheme, which is obtained by replacing each of the (K, α)-coherent matrices by a random selection of its row blocks. This compressed sensing scheme satisfies all four properties (P1)–(P4) simultaneously; to the author's knowledge, the only other scheme to do so is [Iwe14, Theorem 5 (2)]. Both the row count and the recovery algorithm runtime are $O(k \log k \cdot \log N)$. In fact, our scheme is as fast as that of [Iwe14, Theorem 5 (3)], which has the fastest known runtime of schemes with an ℓ_p/ℓ_q error-guarantee, yet the entropy is reduced from $O\left(Nk^2 \log N\right)$ to $O\left(\log k \cdot \log\left(k \log N\right)\right)$. Furthermore, this entropy is as low as that of [Iwe14, Theorem 5 (2)], which is the only previously known scheme satisfying (P1)–(P3) and requiring sublinear entropy. This reduction is a consequence of subsampling row blocks from a Porat-Rothschild matrix $\mathcal{M}_{\mathrm{PR}}$ (see Section 4.1.1)

5.2 Future Work and Open Problems

Trade-off between row count and runtime of recovery algorithm. A nonuniform compressed sensing scheme which achieves an ℓ_1/ℓ_1, ℓ_2/ℓ_1, or ℓ_2/ℓ_2 error guarantee must have a growth rate of $\Omega(k \log(N/k))$ for both the row count of the measurement matrix $\mathcal{M}$ and (unless the measurement matrix is trivial) the runtime of the recovery algorithm (see Section 1.2).

As shown in Table 1.1, the scheme of [NS19, Theorem 1.2] achieves an order-optimal row count of $O(k \log N/k)$, but its runtime is $O(\log N/k)$ times larger than the lower bound. In contrast, the row count and the runtime of the schemes in Corollary 4.4.1 and [Iwe14, Theorem 5 (3)] are both $O(k \log k \cdot \log N)$. This leads to two related questions:

1. Are there nonuniform compressed sensing schemes with an ℓ_1/ℓ_1, ℓ_2/ℓ_1, or ℓ_2/ℓ_2 error guarantee for which both the row count and the runtime are order-optimal?
2. Are there inherent trade-offs between the row count and the runtime?

Trade-off between recovery models. Similar trade-off questions also apply when comparing the nonuniform recovery model with the uniform recovery model or the deterministic recovery model. Recall that the uniform recovery model is stronger than the nonuniform recovery model, because a scheme satisfying an ℓ_p/ℓ_q error guarantee in the former model satisfies it in the latter model too. For the ℓ_2/ℓ_2 error-guarantee, the uniform recovery model requires strictly more resources than the nonuniform recovery model: while there are nonuniform ℓ_2/ℓ_2 schemes with an order-optimal row count of $O(k \log(N/k))$ (see Table 1.1), the row count of a uniform ℓ_2/ℓ_2 scheme must be $\Omega(N)$ [CDD09, Theorem 5.1]. Furthermore, there is no known (nontrivial) scheme satisfying an ℓ_p/ℓ_q error guarantee in the uniform recovery model while having runtime $O(k \text{ polylog } N)$. It is an open problem whether this barrier can be broken. Finally, there is no known deterministic scheme satisfying an ℓ_p/ℓ_q error guarantee while having a row count $O(k \text{ polylog } N)$.

Compressed sensing using generative priors. Inspired by the tremendous success of deep generative models [KW13, GPAM+14, HH19, RH21] in a variety of applications, a new approach to compressed sensing was recently introduced. Rather than assuming the underlying signal to be compressible, we assume instead that it is well-modelled by a deep generative model [BJPD17]. The impressive empirical performance of this approach has motivated significant research to develop a better theoretical understanding; see [OJM+20, §4.2.1] for a detailed summary of recent developments. This approach is still in its infancy, and does not yet include aspects of standard compressed sensing such as the design of deterministic schemes and sublinear-time recovery algorithms.

Bibliography

[AHSC09] Lorne Applebaum, Stephen D. Howard, Stephen Searle, and Robert Calderbank. Chirp sensing codes: Deterministic compressed sensing measurements for fast recovery. Applied and Computational Harmonic Analysis, 26(2):283–290, 2009. doi: 10.1016/j.acha.2008.08.002. (Cited on pages 12 and 14.)

[AJS19] Matthew Aldridge, Oliver Johnson, and Jonathan Scarlett. Group Testing: An Information Theory Perspective. Foundations and Trends® in Communications and Information Theory, 15(3-4):196–392, 2019. doi: 10.1561/0100000099. (Cited on page 16.)

[AM11] Arash Amini and Farokh Marvasti. Deterministic Construction of Binary, Bipolar, and Ternary Compressed Sensing Matrices. IEEE Transactions on Information Theory, 57(4):2360–2370, 2011. doi: 10.1109/TIT.2011.2111670. (Cited on page 12.)

[AMM12] Arash Amini, Vahid Montazerhodjat, and Farokh Marvasti. Matrices With Small Coherence Using p-Ary Block Codes . IEEE Transactions on Signal Processing, 60(1):172, 2012. doi: 10.1109/TSP.2011.2169249. (Cited on pages 12 and 14.)

[AS16] Noga Alon and Joel H. Spencer. The Probabilistic Method. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, Fourth edition, 2016. (Cited on page 15.)

[Bar07] Richard G. Baraniuk. Compressive Sensing. IEEE Signal Processing Magazine, 24(4):118–121, 2007. doi: 10.1109/MSP.2007.4286571. (Cited on page 2.)

[BD09] Thomas Blumensath and Mike E. Davies. Iterative hard thresholding for compressed sensing. Applied and Computational Harmonic Analysis, 27(3):265–274, 2009. doi: 10.1016/j.acha.2009.04.002. (Cited on page 12.)

[BDF$^+$11] Jean Bourgain, Stephen Dilworth, Kevin Ford, Sergei Konyagin, and Denka Kutzarova. Explicit constructions of RIP matrices and related problems. Duke Mathematical Journal, 159(1):145–185, 2011. doi: 10.1215/00127094-1384809. (Cited on pages 7 and 12.)

[BDMS13] Afonso S. Bandeira, Edgar Dobriban, Dustin G. Mixon, and William F. Sawin. Certifying the Restricted Isometry Property is Hard. IEEE Transactions on Information Theory, 59(6):3448–3450, 2013. doi: 10.1109/TIT.2013.2248414. (Cited on page 13.)

[BFP$^+$73] Manuel Blum, Robert W. Floyd, Vaughan Pratt, Ronald L. Rivest, and Robert E. Tarjan. Time bounds for selection. Journal of Computer and System Sciences, 7(4):448–461, 1973. doi: 10.1016/S0022-0000(73)80033-9. (Cited on pages 34 and 57.)

[BIPW10] Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower Bounds for Sparse Recovery. In Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1190–1197, 2010. doi: 10.1137/1.9781611973075.95. (Cited on pages 6 and 12.)

[BIS12] J. Bailey, M. A Iwen, and C. V. Spencer. On the Design of Deterministic Matrices for Fast Recovery of Fourier Compressible Functions. SIAM Journal on Matrix Analysis and Applications, 33(1):263–289, 2012. doi: 10.1137/110835864. (Cited on pages 23, 24, 33, 42, 45, and 62.)

[BJPD17] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed Sensing using Generative Models. In Proceedings of the 34th International Conference on Machine Learning, pages 537–546, 2017. Available at http://proceedings.mlr.press/v70/bora17a.html. (Cited on page 63.)

[BLM18] Afonso S. Bandeira, Megan E. Lewis, and Dustin G. Mixon. Discrete Uncertainty Principles and Sparse Signal Processing. Journal of Fourier Analysis and Applications, 24(4):935–956, 2018. doi: 10.1007/s00041-017-9550-x. (Cited on page 12.)

[Bou14] Jean Bourgain. An Improved Estimate in the Restricted Isometry Problem. In Bo'az Klartag and Emanuel Milman (Editors), Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 2011-2013, pages 65–70. Springer International Publishing, 2014. doi: 10.1007/978-3-319-09477-9_5. (Cited on page 12.)

[Can06] Emmanuel J. Candès. Compressive sampling. In Proceedings of the International Congress of Mathematicians Madrid, August 22–30, 2006, pages 1433–1452, 2006. doi: 10.4171/022-3/69. (Cited on page 2.)

[Can08] Emmanuel J Candés. The restricted isometry property and its implications for compressed sensing. Comptes Rendus Mathematique , 346(9-10):589–592, 2008. doi: 10.1016/j.crma.2008.03.014. (Cited on page 12.)

[CCJS11] Chun Lam Chan, Pak Hou Che, Sidharth Jaggi, and Venkatesh Saligrama. Non-adaptive probabilistic group testing with noisy measurements: Near-optimal bounds with efficient algorithms. In 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pages 1832–1839. IEEE, 2011. doi: 10.1109/Allerton.2011.6120391. (Cited on pages 17 and 18.)

[CDD09] Albert Cohen, Wolfgang Dahmen, and Ronald DeVore. Compressed sensing and best k-term approximation. Journal of the American mathematical society, 22(1):211–231, 2009. doi: 10.1090/S0894-0347-08-00610-3. (Cited on pages 4 and 63.)

[Che11a] Mahdi Cheraghchi. Applications of Derandomization Theory in Coding. PhD thesis, École polytechnique fédérale de Lausanne (EPFL), 2011. doi: 10.5075/epfl-thesis-4767. (Cited on page 18.)

[Che11b] Mahdi Cheraghchi. Coding-Theoretic Methods for Sparse Recovery. In 2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pages 909–916. IEEE, 2011. doi: 10.1109/Allerton.2011.6120263. (Cited on pages 12, 18, 21, and 22.)

[CJ10] Robert Calderbank and Sina Jafarpour. Reed Muller Sensing Matrices and the LASSO. In Claude Carlet and Alexander Pott (Editors), International Conference on Sequences and Their Applications, pages 442–463. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-15874-2_37. (Cited on page 12.)

[CJSA14] Chun Lam Chan, Sidharth Jaggi, Venkatesh Saligrama, and Samar Agnihotri. Non-Adaptive Group Testing: Explicit Bounds and Novel Algorithms. IEEE Transactions on Information Theory, 60(5):3019–3035, 2014. doi: 10.1109/TIT.2014.2310477. (Cited on page 17.)

[CM06] Graham Cormode and S. Muthukrishnan. Combinatorial Algorithms for Compressed Sensing. In Structural Information and Communication Complexity, pages 280–294. Springer Berlin Heidelberg, 2006. doi: 10.1007/11780823_22. (Cited on pages 7, 11, and 14.)

[CMS03] Andrea E.F. Clementi, Angelo Monti, and Riccardo Silvestri. Distributed broadcast in radio networks of unknown topology. Theoretical Computer Science, 302(1-3):337–364, 2003. doi: 10.1016/S0304-3975(02)00851-4. (Cited on page 17.)

[CN20] Mahdi Cheraghchi and Vasileios Nakos. Combinatorial Group Testing and Sparse Recovery Schemes with Near-Optimal Decoding Time. 2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS), pages 1203–1213, 2020. doi: 10.1109/FOCS46700.2020.00115. (Cited on page 17.)

[CRT06a] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information. IEEE Transactions on Information Theory, 52(2):489–509, 2006. doi: 10.1109/TIT.2005.862083. (Cited on pages 2, 10, and 12.)

[CRT06b] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable Signal Recovery from Incomplete and Inaccurate Measurements. Communications on Pure and Applied Mathematics, 59(8):1207–1223, 2006. doi: 10.1002/cpa.20124. (Cited on page 2.)

[CT05] Emmanuel J. Candes and Terence Tao. Decoding by Linear Programming. IEEE Transactions on Information Theory, 51(12):4203–4215, 2005. doi: 10.1109/TIT.2005.858979. (Cited on page 2.)

[CT06a] Emmanuel J. Candes and Terence Tao. Near-Optimal Signal Recovery From Random Projections: Universal Encoding Strategies?

IEEE Transactions on Information Theory, 52(12):5406–5425, 2006. doi: 10.1109/TIT.2006.885507. (Cited on pages 2 and 12.)

[CT06b] Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. John Wiley & Sons, Inc., Second edition, 2006. (Cited on page 50.)

[CW08] Emmanuel J. Candès and Michael B. Wakin. An Introduction To Compressive Sampling. IEEE Signal Processing Magazine, 25(2):21–30, 2008. doi: 10.1109/MSP.2007.914731. (Cited on page 2.)

[DE03] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization. Proceedings of the National Academy of Sciences, 100(5):2197–2202, 2003. doi: 10.1073/pnas.0437847100. (Cited on page 12.)

[DE11] Marco F. Duarte and Yonina C. Eldar. Structured compressed sensing: From theory to applications. IEEE Transactions on Signal Processing, 59(9):4053–4085, 2011. doi: 10.1109/TSP.2011.2161982. (Cited on page 5.)

[DeV07] Ronald A. DeVore. Deterministic constructions of compressed sensing matrices. Journal of Complexity, 23(4):918–925, 2007. doi: 10.1016/j.jco.2007.04.002. (Cited on pages 12, 13, and 14.)

[DH00] Ding-Zhu Du and Frank Hwang. Combinatorial Group Testing and Its Applications, volume 12. World Scientific, 2nd edition, 2000. doi: 10.1142/4252. (Cited on pages 16 and 19.)

[DH06] Ding-Zhu Du and Frank Hwang. Pooling Designs and Nonadaptive Group Testing. World Scientific, 2006. doi: 10.1142/6122. (Cited on page 16.)

[DKWB20] Yunzi Ding, Dmitriy Kunisky, Alexander S. Wein, and Afonso S. Bandeira. The Average-Case Time Complexity of Certifying the Restricted Isometry Property. arXiv: 2005.11270v3 [math.ST], 2020. (Cited on page 13.)

[Don06] David L. Donoho. Compressed Sensing. IEEE Transactions on Information Theory, 52(4):1289–1306, 2006. doi: 10.1109/TIT.2006.871582. (Cited on pages 2, 10, and 12.)

[Don18] David Donoho. FROM BLACKBOARD TO BEDSIDE - GAUSS PRIZE LECTURE, pages 211–224. World Scientific, 2018. doi: 10.1142/9789813272880_0010. (Cited on pages 2 and 5.)

[DR82] A.G. D'yachkov and V.V. Rykov. Bounds on the length of disjunctive codes. Problemy Peredachi Informatsii, 18(3):7–13, 1982. Translation: Problems of Information Transmission. 18(3): 166–171. Available at https://www.researchgate.net/publication/235008674_Survey_of_Superimposed_Code_Theory. (Cited on pages 16, 17, and 18.)

[DSV12] Alexandros G. Dimakis, Roxana Smarandache, and Pascal O. Vontobel. LDPC Codes for Compressed Sensing. IEEE Transactions on Information Theory, 58(5):3093–3114, 2012. doi: 10.1109/TIT.2011.2181819. (Cited on page 14.)

[EFF85] P. Erdös, P. Frankl, and Z. Füredi. Families of finite sets in which no set is covered by the union of r others. Israel Journal of Mathematics, 51(1-2):79–89, 1985. doi: 10.1007/BF02772959. (Cited on page 17.)

[EK12] Yonina C. Eldar and Gitta Kutyniok (Editors). Compressed Sensing: Theory and Applications. Cambridge University Press, 2012. doi: 10.1017/CBO9780511794308. (Cited on page 2.)

[FBB$^+$17] Li Feng, Thomas Benkert, Kai Tobias Block, Daniel K Sodickson, Ricardo Otazo, and Hersh Chandarana. Compressed Sensing for Body MRI. Journal of Magnetic Resonance Imaging, 45(4):966–987, 2017. doi: 10.1002/jmri.25547. (Cited on page 5.)

[For65] George David Forney. Concatenated Codes. PhD thesis, Massachusetts Institute of Technology, 1965. Available at https://dspace.mit.edu/handle/1721.1/13449. (Cited on page 16.)

[FR13] Simon Foucart and Holger Rauhut. A Mathematical Introduction to Compressive Sensing. Birkhäuser Basel, 2013. doi: 10.1007/978-0-8176-4948-7. (Cited on pages 2, 10, and 27.)

[Fü96] Zoltán Füredi. On r-cover-free families. Journal of Combinatorial Theory, Series A, 73(1):172 – 173, 1996. doi: 10.1006/jcta.1996.0012. (Cited on page 17.)

[GI10] Anna Gilbert and Piotr Indyk. Sparse Recovery Using Sparse Matrices. Proceedings of the IEEE, 98(6):937–947, 2010. doi: 10.1109/JPROC.2010.2045092. (Cited on page 11.)

[Gil52] E.N. Gilbert. A comparison of signalling alphabets. The Bell System Technical Journal, 31(3):504–522, 1952. doi: 10.1002/j.1538-7305.1952.tb01393.x. (Cited on page 15.)

[GIS08] Anna C. Gilbert, Mark A. Iwen, and Martin J. Strauss. Group testing and sparse signal recovery. In 2008 42nd Asilomar Conference on Signals, Systems and Computers, pages 1059–1063. IEEE, 2008. doi: 10.1109/ACSSC.2008.5074574. (Cited on page 18.)

[GLPS12] Anna C. Gilbert, Yi Li, Ely Porat, and Martin J. Strauss. Approximate Sparse Recovery: Optimizing Time and Measurements. SIAM Journal on Computing, 41(2):436–453, 2012. doi: 10.1137/100816705. (Cited on page 7.)

[GLPS17] Anna C. Gilbert, Yi Li, Ely Porat, and Martin J Strauss. For-All Sparse Recovery in Near-Optimal Time. ACM Transactions on Algorithms, 13(3):1–26, 2017. doi: 10.1145/3039872. (Cited on page 7.)

[GPAM$^+$14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Nets. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press. Available at https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html. (Cited on page 63.)

[GRS19] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential Coding Theory, 2019. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/index.html. (Cited on page 16.)

[GSTV07] A. C. Gilbert, M. J. Strauss, J. A. Tropp, and R. Vershynin. One Sketch for All: Fast Algorithms for Compressed Sensing. In STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, pages 237–246, 2007. doi: 10.1145/1250790.1250824. (Cited on page 7.)

[HCS08] S. D. Howard, A. R. Calderbank, and S. J. Searle. A Fast Reconstruction Algorithm for Deterministic Compressive Sensing using Second Order Reed-Muller Codes. In 2008 42nd Annual Conference on Information Sciences and Systems, pages 11–15. IEEE, 2008. doi: 10.1109/CISS.2008.4558486. (Cited on page 14.)

[HH19] Catherine F. Higham and Desmond J. Higham. Deep Learning: An Introduction for Applied Mathematicians. SIAM Review, 61(4):860–891, 2019. doi: 10.1137/18M1165748. (Cited on page 63.)

[HHL10] Justin P Haldar, Diego Hernando, and Zhi-Pei Liang. Compressed-Sensing MRI With Random Encoding. IEEE Transactions on Medical Imaging, 30(4):893–903, 2010. doi: 10.1109/TMI.2010.2085084. (Cited on page 5.)

[HR17] Ishay Haviv and Oded Regev. The Restricted Isometry Property of Subsampled Fourier Matrices. In Bo'az Klartag and Emanuel Milman (Editors), Geometric Aspects of Functional Analysis: Israel Seminar (GAFA) 2014-2016, pages 163–179. Springer, 2017. doi: 10.1007/978-3-319-45282-1_11. (Cited on pages 12 and 13.)

[HS09] Matthew A Herman and Thomas Strohmer. High-Resolution Radar via Compressed Sensing. IEEE Transactions on Signal Processing, 57(6):2275–2284, 2009. doi: 10.1109/TSP.2009.2014277. (Cited on page 12.)

[Ind08] Piotr Indyk. Explicit Constructions for Compressed Sensing of Sparse Signals. In SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, volume 8, pages 30–33, 2008. Available at https://dl.acm.org/doi/proceedings/10.5555/1347082?id=61. (Cited on page 11.)

[INR10] Piotr Indyk, Hung Q. Ngo, and Atri Rudra. Efficiently Decodable Non-adaptive Group Testing. In Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1126–1142. SIAM, 2010. doi: 10.1137/1.9781611973075.91. (Cited on page 17.)

[Iwe14] M.A. Iwen. Compressed sensing with sparse binary matrices: Instance optimal error guarantees in near-optimal time. Journal of Complexity, 30(1):1 – 15, 2014. doi: 10.1016/j.jco.2013.08.001. (Cited on pages 7, 8, 46, 47, 50, 52, 55, 59, 61, 62, and 63.)

[JMF13] John Jasper, Dustin G. Mixon, and Matthew Fickus. Kirkman Equiangular Tight Frames and Codes. IEEE Transactions on Information Theory, 60(1):170–181, 2013. doi: 10.1109/TIT.2013.2285565. (Cited on page 12.)

[Joh62] Selmer M. Johnson. A New Upper Bound for Error-Correcting Codes. IRE Transactions on Information Theory, 8(3):203–207, 1962. doi: 10.1109/TIT.1962.1057714. (Cited on page 19.)

[Kat73] G.O.H. Katona. Combinatorial Search Problems. In J. N. Srivastava (Editor), A Survey of Combinatorial Theory, pages 285–308. North-Holland, 1973. doi: 10.1016/B978-0-7204-2262-7.50028-4. (Cited on page 16.)

[KN10] Daniel M Kane and Jelani Nelson. A Derandomized Sparse Johnson-Lindenstrauss Transform. arXiv: 1006.3585v3 [cs.DS], 2010. (Cited on page 7.)

[KS64] W. Kautz and R. C. Singleton. Nonrandom Binary Superimposed Codes . IEEE Transactions on Information Theory, 10(4):363–377, 1964. doi: 10.1109/TIT.1964.1053689. (Cited on pages 8, 16, 17, 20, 21, and 62.)

[KW13] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. arXiv: 1312.6114v10 [stat.ML], 2013. (Cited on page 63.)

[LDP07] Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The Application of Compressed Sensingfor Rapid MR Imaging. Magnetic Resonance in Medicine, 58(6):1182–1195, 2007. doi: 10.1002/mrm.21391. (Cited on page 5.)

[LDSP08] Michael Lustig, David L Donoho, Juan M Santos, and John M Pauly. Compressed Sensing MRI. IEEE Signal Processing Magazine, 25(2):72–82, 2008. doi: 10.1109/MSP.2007.914728. (Cited on page 5.)

[LG08] Jun Luo and Dongning Guo. Neighbor Discovery in Wireless Ad Hoc Networks Based on Group Testing. In 2008 46th Annual Allerton Conference on Communication, Control, and Computing, pages 791–797, 2008. doi: 10.1109/ALLERTON.2008.4797638. (Cited on page 17.)

[LG14a] Shuxing Li and Gennian Ge. Deterministic Construction of Sparse Sensing Matrices via Finite Geometry. IEEE Transactions on Signal Processing, 62(11):2850–2859, 2014. doi: 10.1109/TSP.2014.2318139. (Cited on page 12.)

[LG14b] Shuxing Li and Gennian Ge. Deterministic Sensing Matrices Arising From Near Orthogonal Systems. IEEE Transactions on Information Theory, 60(4):2291–2302, 2014. doi: 10.1109/TIT.2014.2303973. (Cited on page 12.)

[LGGZ12] Shuxing Li, Fei Gao, Gennian Ge, and Shengyuan Zhang. Deterministic Construction of Compressed Sensing Matrices via Algebraic Curves. IEEE Transactions on Information Theory, 58(8):5035–5041, 2012. doi: 10.1109/TIT.2012.2196256. (Cited on pages 12 and 14.)

[Li13] Yi Li. Sublinear Time Algorithms for the Sparse Recovery Problem. PhD thesis, University of Michigan, 2013. Available at https://hdl.handle.net/2027.42/102438. (Cited on page 11.)

[LJ21] Ivan Lau and Jonathan Jedwab. Construction of binary matrices for near-optimal compressed sensing, 2021. To appear in 2021 IEEE International Symposium on Information Theory (ISIT), pages 1612–1617. (Cited on page 9.)

[LV20] Mahsa Lotfi and Mathukumalli Vidyasagar. Compressed Sensing Using Binary Matrices of Nearly Optimal Dimensions. IEEE Transactions on Signal Processing, 68:3008–3021, 2020. doi: 10.1109/TSP.2020.2990154. (Cited on page 14.)

[MAD$^+$12] Mark Murphy, Marcus Alley, James Demmel, Kurt Keutzer, Shreyas Vasanawala, and Michael Lustig. Fast ℓ_1-SPIRiT Compressed Sensing Parallel Imaging MRI: Scalable Parallel Implementation and Clinically Feasible Runtime. IEEE Transactions on Medical Imaging, 31(6):1250–1262, 2012. doi: 10.1109/TMI.2012.2188039. (Cited on page 5.)

[Mal13] Mikhail Malyutov. Search for Sparse Active Inputs: A Review. In Harout Aydinian, Ferdinando Cicalese, and Christian Deppe (Editors), Information Theory, Combinatorics, and Search Theory: In Memory of Rudolf Ahlswede, page 609–647. Springer-Verlag, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-36899-8_31. (Cited on page 17.)

[MB11] Arya Mazumdar and Alexander Barg. General Constructions of Deterministic (S)RIP Matrices for Compressive Sampling. In 2011 IEEE International Symposium on Information Theory Proceedings, pages 678–682. IEEE, 2011. doi: 10.1109/ISIT.2011.6034217. (Cited on page 14.)

[Mix15] Dustin G. Mixon. Explicit Matrices with the Restricted Isometry Property: Breaking the Square-Root Bottleneck. In Holger Boche, Robert Calderbank, Gitta Kutyniok, and Jan Vybíral (Editors), Compressed Sensing and its Applications: : MATHEON Workshop 2013, pages 389–417. Springer International Publishing, 2015. doi: 10.1007/978-3-319-16042-9_13. (Cited on pages 7 and 12.)

[MSW$^+$21] John H. McDermott, Duncan Stoddard, Peter J. Woolf, Jamie M. Ellingford, David Gokhale, Algy Taylor, Leigh A.M. Demain, William G. Newman, and Graeme Black. A Nonadaptive Combinatorial Group Testing Strategy to Facilitate Health Care Worker Screening during the Severe Acute Respiratory Syndrome Coronavirus-2 (SARS-CoV-2) Outbreak. The Journal of Molecular Diagnostics, 2021. doi: 10.1016/j.jmoldx.2021.01.010. (Cited on page 17.)

[Mut06] S Muthukrishnan. Some Algorithmic Problems and Results in Compressed Sensing. In Allerton Conference, 2006. Available at http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=D641EDC91AC401B75CD32FEFDEAAD75B?doi=10.1.1.90.6581&rep=rep1&type=pdf. (Cited on page 11.)

[Nal20] Ozkan Ufuk Nalbantoglu. Group testing performance evaluation for SARS-CoV-2 massive scale screening and testing. BMC Medical Research Methodology, 20(176), 2020. doi: 10.1186/s12874-020-01048-1. (Cited on page 17.)

[NJS16] R. Ramu Naidu, Phanindra Jampana, and C. S. Sastry. Deterministic Compressed Sensing Matrices: Construction via Euler Squares and Applications. IEEE Transactions on Signal Processing, 64(14):3566–3575, 2016. doi: 10.1109/TSP.2016.2550020. (Cited on page 12.)

[NPR11] Hung Q Ngo, Ely Porat, and Atri Rudra. Efficiently Decodable Error-Correcting List Disjunct Matrices and Applications. In Luca Aceto, Monika Henzinger, and Jiří Sgall (Editors), Automata, Languages and Programming, pages 557–568. Springer Berlin Heidelberg, 2011. (Cited on page 17.)

[NPW14] Jelani Nelson, Eric Price, and Mary Wootters. New constructions of RIP matrices with fast multiplication and fewer rows. In Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 1515–1528. SIAM, 2014. doi: 10.1137/1.9781611973402.111. (Cited on page 12.)

[NS19] Vasileios Nakos and Zhao Song. Stronger L2/L2 Compressed Sensing; Without Iterating. In STOC 2019: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, pages 289–297, 2019. doi: 10.1145/3313276.3316355. (Cited on pages 7 and 63.)

[NT09] D. Needell and J.A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. Applied and Computational Harmonic Analysis, 26(3):301–321, 2009. doi: 10.1016/j.acha.2008.07.002. (Cited on page 12.)

[NT11] J.L. Nelson and V.N. Temlyakov. On the size of incoherent systems. Journal of Approximation Theory, 163(9):1238–1245, 2011. doi: 10.1016/j.jat.2011.04.001. (Cited on page 12.)

[NV09] Deanna Needell and Roman Vershynin. Uniform Uncertainty Principle and Signal Recovery via Regularized Orthogonal Matching Pursuit. Foundations of Computational Mathematics, 9(3):317–334, 2009. doi: 10.1007/s10208-008-9031-3. (Cited on page 12.)

[NV10] Deanna Needell and Roman Vershynin. Signal Recovery From Incomplete and Inaccurate Measurements Via Regularized Orthogonal Matching Pursuit. IEEE Journal of Selected Topics in Signal Processing, 4(2):310–316, 2010. doi: 10.1109/JSTSP.2010.2042412. (Cited on page 12.)

[OJM+20] Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep Learning Techniques for Inverse Problems in Imaging. IEEE Journal on Selected Areas in Information Theory, 2020. doi: 10.1109/JSAIT.2020.2991563. (Cited on page 63.)

[PR11] Ely Porat and Amir Rothschild. Explicit Nonadaptive Combinatorial Group Testing Schemes. IEEE Transactions on Information Theory, 57(12):7982–7989, 2011. doi: 10.1109/TIT.2011.2163296. (Cited on pages 8, 15, 18, and 62.)

[PW11] Eric Price and David P. Woodruff. (1+ ϵ)-approximate Sparse Recovery. In 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, pages 295–304. IEEE, 2011. doi: 10.1109/FOCS.2011.92. (Cited on pages 6 and 12.)

[RH21] Lars Ruthotto and Eldad Haber. An introduction to deep generative modeling. GAMM Mitteilungen, 2021. doi: 10.1002/gamm.202100008. (Cited on page 63.)

[Ric] Rice's Compressive Sensing Resources. http://dsp.rice.edu/cs/. Accessed: 08 July 2021. (Cited on page 2.)

[Rom08] Justin Romberg. Imaging via Compressive Sampling. IEEE Signal Processing Magazine, 25(2):14–20, 2008. doi: 10.1109/MSP.2007.914729. (Cited on page 2.)

[Rot06] Ron Roth. Introduction to Coding Theory. Cambridge University Press, 2006. doi: 10.1017/CBO9780511808968. (Cited on pages 14 and 15.)

[RS60] I. S. Reed and G. Solomon. Polynomial Codes Over Certain Finite Fields. Journal of the Society for Industrial and Applied Mathematics, 8(2):300–304, 1960. doi: 10.1137/0108018. (Cited on pages 15 and 20.)

[Rus94] Miklós Ruszinkó. On the upper bound of the size of the r-cover-free families. Journal of Combinatorial Theory, Series A, 66(2):302 – 310, 1994. doi: 10.1016/0097-3165(94)90067-1. (Cited on page 17.)

[RV08] Mark Rudelson and Roman Vershynin. On Sparse Reconstruction from Fourier and Gaussian Measurements. Communications on Pure and Applied Mathematics, 61(8):1025–1045, 2008. doi: 10.1002/cpa.20227. (Cited on page 12.)

[Seo20] Jin-Taek Seong. Group Testing-Based Robust Algorithm for Diagnosis of COVID-19. Diagnostics, 10(6):396, 2020. doi: 10.3390/diagnostics10060396. (Cited on page 17.)

[Sin64] Richard C. Singleton. Maximum Distance Q-Nary Codes. IEEE Transactions on Information Theory, 10(2):116–118, 1964. doi: 10.1109/TIT.1964.1053661. (Cited on page 14.)

[Täu20] Matthias Täufer. Rapid, large-scale, and effective detection of COVID-19 via non-adaptive testing. Journal of Theoretical Biology, 506, 2020. doi: 10.1016/j.jtbi.2020.110450. (Cited on page 17.)

[TG07] Joel A. Tropp and Anna C. Gilbert. Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit. IEEE Transactions on Information Theory, 53(12):4655–4666, 2007. doi: 10.1109/TIT.2007.909108. (Cited on page 18.)

[Til15] Andreas M. Tillmann. Equivalence of linear programming and basis pursuit. Proceedings in Applied Mathematics and Mechanics, 15(1):735–738, 2015. doi: 10.1002/pamm.201510351. (Cited on page 10.)

[TP13] Andreas M. Tillmann and Marc E. Pfetsch. The Computational Complexity of the Restricted Isometry Property, the Nullspace Property, and Related Concepts in Compressed Sensing. IEEE Transactions on Information Theory, 60(2):1248–1259, 2013. doi: 10.1109/TIT.2013.2290112. (Cited on page 13.)

[Var57] R. R. Varshamov. Estimate of the number of signals in error correcting codes (Russian). Doklady Akademii Nauk SSSR, 117:739–741, 1957. (Cited on page 15.)

[VFH$^+$20] Claudio M. Verdun, Tim Fuchs, Pavol Harar, Dennis Elbrächter, David S. Fischer, Julius Berner, Philipp Grohs, Fabian J. Theis, and Felix Krahmer. Group testing for SARS-CoV-2 allows for up to 10-fold efficiency increase across realistic scenarios and testing strategies. medRxiv, 2020. doi: 10.1101/2020.04.30.20085290. (Cited on page 17.)

[WBP16] Tengyao Wang, Quentin Berthet, and Yaniv Plan. Average-case hardness of RIP certification. In Advances in Neural Information Processing Systems 29 (NIPS 2016), pages 3819–3827, 2016. Available at https://papers.nips.cc/paper/2016/hash/d54e99a6c03704e95e6965532dec148b-Abstract.html. (Cited on page 13.)

[Wee17] Jonathan Weed. Approximately Certifying the Restricted Isometry Property is Hard. IEEE Transactions on Information Theory, 64(8):5488–5497, 2017. doi: 10.1109/TIT.2017.2776131. (Cited on page 13.)

[Wel74] L. R. Welch. Lower Bounds on the Maximum Cross Correlation of Signals. IEEE Transactions on Information theory, 20(3):397–399, 1974. doi: 10.1109/TIT.1974.1055219. (Cited on page 12.)

[YZ13] Nam Yul Yu and Na Zhao. Deterministic Construction of Real-Valued Ternary Sensing Matrices Using Optical Orthogonal Codes. IEEE Signal Processing Letters, 20(11):1106–1109, 2013. doi: 10.1109/LSP.2013.2281597. (Cited on pages 12 and 14.)

[ZDL$^+$21] Hooman Zabeti, Nick Dexter, Ivan Lau, Leonhardt Unruh, Ben Adcock, and Leonid Chindelevitch. Group Testing Large Populations for SARS-CoV-2. medRxiv, 2021. doi: 10.1101/2021.06.03.21258258. (Cited on page 17.)